mc

September 1988
7 DM · 60 öS · 7 sFr

Die Mikrocomputer-Zeitschrift

9/88

Compaq und IBM stellen vor:

Neue PC-Generation

Transputer-Power für PC

Speicherverwaltung verstehen

Atari ST: Hardcopy verbessern

C-Workshop: Header-Dateien automatisch

Im Test:

- Modula-2-Systeme
- Paralleles C für die mc-Transputerkarte
- Platinen entwerfen mit PC
- Digitalisiertabletts

DOS-Workshop: Dateiaufbereitung für dBase III plus

VORSPRUNG MIT PLANTRON

Die neue PLANTRON PT-AT TOWER Serie! Mit der bekannten Komplettausstattung – Führend in Design und Technik.

- CPU 80286, 8/10 MHz schaltbar (0/1 Wait-State)
- 640 KB RAM (bis 1 MB on Board 640/384 KB)
- Parallele, serielle Schnittstelle und Game Port
- 1.2 MB Diskettenlaufwerk (5.25")
- 720 KB Diskettenlaufwerk (3.5") optional
- 64 MB (netto) Festplattenlaufwerk optional
- Monochrom-Grafikkarte 720x348 Punkte
- Ultra-EGA-Karte 800x600 Punkte optional
- Große DIN-Tastatur (abschließbar)
- FTZ-Zulassung
- MS-DOS 3.3 (deutsch) mit GW-BASIC optional

ab DM 3498.–
(ohne Monitor)

Alle Computersysteme werden mit ausführlichem deutschen Handbuch geliefert.
PLANTRON-Produkte erhalten Sie nur im autorisierten Fachhandel. Bitte fordern Sie Unterlagen zu unserem Gesamtlieferprogramm vom PT-XT bis zum PT-386 sowie das Fachhändlerverzeichnis an.

Technische Änderungen vorbehalten
© 1988 by PLANTRON GmbH (Europe)

PLANTRON
Computer Vertriebsgesellschaft mbH
Höhestraße 28 · D-6380 Bad Homburg v.d.H. · Telefon (06172) 81031* · Telefax (06172) 81036
Telex 417410 placo d

R & H. Flöhner GmbH Visuelle Kommunikation Grafik Design BDW 6382 Friedrichsdorf/Ts.

EDITORIAL

Wir berichten in jeder Ausgabe aktuell über technische Entwicklungen, Neuheiten und Trends aus der Mikrocomputer-Szene.
Immer kürzer wird die Zeit, die Hard- und Software-Anbieter brauchen, um noch leistungsfähigere Systeme auf den Markt zu werfen. Da ist es von Vorteil, fundiert über die Entwicklung informiert zu sein. Nur dann kann man eine Entscheidung für ein Produkt reifen lassen. Wer vor zwei Monaten z. B. etwas über eine Grafikkarte mit bislang unerreichten Leistungsdaten gelesen hat, findet in dieser Ausgabe eine noch bessere. Vermutlich wird es in wenigen Monaten eine noch bessere geben ...

Liebe Leser,

Eine Produktgruppe, die wir in dieser Ausgabe unter die Lupe der Kritik legen, ist die der Modula-2-Compiler. Sie werden immer häufiger zur professionellen Software-Entwicklung eingesetzt und unterscheiden sich doch erheblich voneinander. In Jochen Guther haben wir für den Vergleich einen professionellen Programmierer und Kritiker gleichermaßen gefunden. Dieser Mann weiß durch langjährige Erfahrung, wovon er spricht und worauf es bei Modula-2-Compilern ankommt.

Neben dem umfangreichen Blick auf das allgemeine Marktgeschehen widmet sich mc den alltäglichen Problemen und Bedürfnissen der Anwender. Sei es die Abfrage der Tastatur aus einem C-Programm heraus oder ein Tip zum effektiveren Einsatz von MS-DOS.
Selbst wer tagtäglich mit seinem Computer arbeitet, kennt längst noch nicht alle Kniffe, die ihm seine Arbeit oft wesentlich erleichtern könnten. Nur wer sich stetig weiterbildet – und dazu gehört auch, Fachzeitschriften wie die mc zu lesen – kann dazulernen und effektiver arbeiten.

Effektiver arbeiten war auch das Motto, unter dem wir uns Grafik-Tabletts für PC angesehen haben. Welche Anforderungen muß man an diese Werkzeuge stellen – und welche erfüllen Sie? In Verbindung mit einem guten CAD-Programm kann der Ingenieur die Phase der Konstruktion und Entwicklung erheblich verkürzen. Auch über ein CAD-Programm finden Sie in dieser Ausgabe einen interessanten Artikel, der von einem Praktiker für Praktiker geschrieben wurde.

Transputer sorgen seit Wochen für interessante Diskussionen. Viele haben davon gehört, einige haben schon einen gesehen und ganz wenige arbeiten tatsächlich damit. mc-Leser wissen durch unsere Transputer-Serie natürlich mehr als die anderen. Für den PC-Anwender ist eine Transputer-Steckkarte genau das richtige, um erste Erfahrungen zu sammeln und gleichzeitig einen Nutzen, nämlich wesentlich schnellere Programme, daraus zu ziehen. Daß diese Karte in mancher Hinsicht selbst eine VAX aussticht, wird in dieser Ausgabe bewiesen.

Bei allem Streben nach dem Besseren, Schöneren und Effektiveren sollte der Blick zurück auf die geschichtliche Entwicklung der Technik nicht unterbleiben. Diesmal berichten wir über die Verfahren zur Speicherverwaltung, durch deren Fortschritt der Einsatz eines Hauptspeichers von mehreren MByte erst einen Sinn macht.

Viel Spaß beim Lesen der mc – und einige Aha-Erlebnisse – wünscht Ihnen

Ihr

Rainer Scholl

DIE UNIC-ICB-IDEE

XT/AT-KOMPATIBLE COMPUTER IN INDUSTRIELLEM DESIGN

Die starke Verbreitung von IBM-kompatiblen Rechnersystemen hat es mit sich gebracht, daß eine sehr große Anzahl von entsprechenden Betriebssystemen, Programmiersprachen, Utilities und Programmgeneratoren rasch und preisgünstig erhältlich ist. Die Verwendung dieser Programme für den industriellen Einsatz (Maschinensteuerung, Betriebsdatenerfassung) scheiterte meist daran, daß normale Rechnersysteme für rauhe Umgebungsbedingungen nicht ausgelegt waren (erschütterungsempfindlich, Probleme mit Staub, Temperatur und Feuchtigkeit).

VORTEILE DER UNIC-ICB-RECHNER

100 % softwarekompatibel zu IBM XT/AT-Rechnern. Vollständiger Aufbau auf Doppel- und Einfach-Europakarten – genormte Gehäuse, beliebige Schutzarten. Erweiterter Temperaturbereich durch entsprechende Steckverbindungen und Bauteileauswahl. Erschütterungsunempfindlich durch Einsatz nichtrotierender Massenspeicher (RAM/EPROM-Floppy).
Vollständig modular aufgebautes System, dessen Komponenten für Softwareentwicklung und eventuell Debugging auch am Personalcomputer angeschlossen werden können. Original IBM-Erweiterungskarten (Netzwerkkarten, verschiedene Schnittstellenkarten usw.) sind über eine Adapterkarte direkt auf ICB-Systemen einsetzbar.
Durch die Verwendungsmöglichkeit eines Personalcomputers zur Programmentwicklung für eine industrielle Applikation entfällt die ansonsten notwendige Anschaffung eines eigenen Entwicklungssystems (VME, ECB usw.).
Durch konsequente 19"-Technik (zum Einsatz kommen Platinen im Einfach- und/oder Doppel-Euro-Format) sind ohne großen Aufwand beliebige Schutzarten nach DIN 40 050 realisierbar.

INDUTRONIC®
Dipl.-Ing. Kreiger GmbH & Co. KG
D-8011 Poing/München
Gruberstraße 46
Telefon (0 81 21) 7 01-0
Telefax (0 81 21) 7 13 64

Ich bitte um Zusendung weiterer Informationen:

Name:

Firma:

Straße:

Plz/Ort:

Telefon:

INHALT

INFO

Messen im Herbst 34

AT- und Modell 30-Einplatinen-Computer von Western Digital 67

Schneller Coprozessor von Weitek 103

PLD-Compiler vereinfacht Entwicklung 106

Extrem schnelle lineare Semicustom-Arrays 135

TEST

Mikrokanal-Duett 42

Mit einem verbesserten Modell 50 und dem neuen Modell 70 rundet IBM die PS/2-Produktpalette ab.

Kompakter AT aus Taiwan 44

Der MPC2000VE von Mitac aus der AT-Klasse übertrifft sogar 80386-Rechner an Geschwindigkeit.

Zwei aus einem Stall 48

Die neuen 80386-Rechner von Compaq: Der eine fast eine Workstation, der andere im Low-Cost-Bereich.

Transputer-Power für den PC 52

Wie man mit einer Steckkarte seinen PC zu Höchstleistungen treiben kann!

CAD-Layout, ein spezieller Markt 54

Das System Draftsman aus dem Low-Cost-Bereich.

Digitalisier-Tabletts aller Größen 62

Wir stellen fünf Tabletts gegenüber.

Titan sucht mit Erfolg 65

Schnelles, unkompliziertes Finden von Text oder Daten in sehr großen Beständen, das sollen Retrievalsysteme leisten.

Modula-2 wird erwachsen 68

Die Entwicklungssysteme von ITC, JPI, Logitech und Stony Brook werden in einem umfangreichen Vergleich auf Herz und Nieren getestet.

Parallelprogrammierung und Transputer-Netzwerke in C 78

Um C tauglich für Transputer zu machen, muß ein Compiler sich erst auf dessen besondere Eigenschaften einstellen.

Low-Cost-CAD-Programm 82

Auf Atari ST und PC (jeweils unter GEM) zeigt GFA Draft Plus, daß gute CAD-Programme nicht teuer sein müssen.

GRUNDLAGEN

Zwischen Textverarbeitung und relationaler Datenbank 66

Zur Stellung und Bedeutung von Retrievalsystemen.

Speicherverwaltungstechniken 136

Die Leistungssteigerung moderner Computer ist neben Verarbeitungsbreite und Taktfrequenz auch in geänderten Organisationsformen bei der Speicherverwaltung zu suchen.

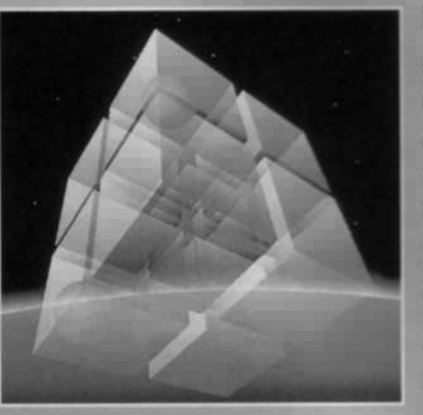
SOFTWARE

EGA-Wonder und Turbo-Pascal 4.0 77

Wie auch dieses Pärchen zueinander findet ...

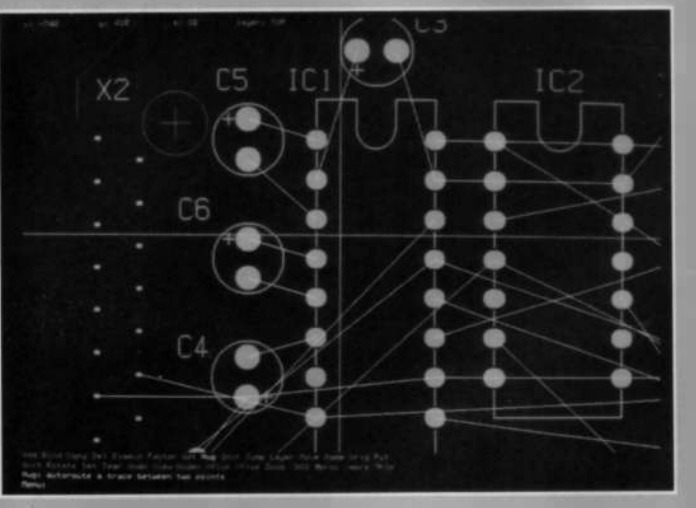
Die Neuen von IBM, Compaq und Mitac

Kurz nach ihrer Vorstellung wurden die neuen Computer von IBM, Compaq und Mitac von mc getestet. Während die PS/2-Serie von IBM um ein neues Modell ergänzt wird, bringt Compaq gleich zwei neue 80386-Systeme heraus. Wie bei Compaq fast schon zu erwarten, setzen sie neue Maßstäbe in der Geschichte des PC. Mitac demonstriert mit seinem Modell, daß auch aus einem 80286-Rechner noch einiges herauszuholen ist.
Die Serie der Rekorde beginnt auf **Seite 42**



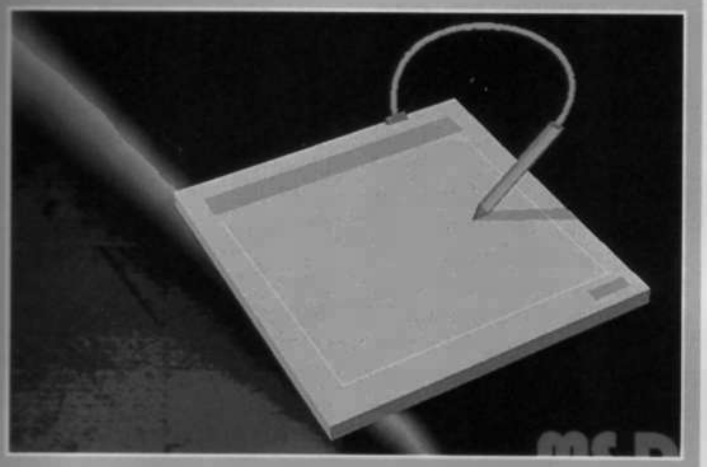
Parallelprogrammierung in C

C ist die Programmiersprache, die sich durchgesetzt hat, wenn man hart an der Hardware entlang programmieren möchte, aber nicht in den unstrukturierten Tiefen spezieller Assembler versinken möchte. Deshalb ist in diesem Heft geschildert, wie der C-Compiler von Cesys mit Parallel-Funktionen erweitert wird:
Eine C-Variante für Transputer. **Seite 78**



CAD-Layout, ein spezieller Markt

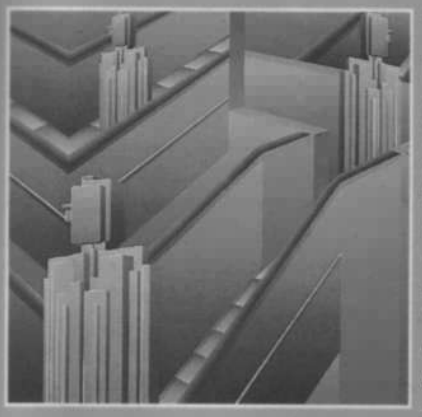
Die neue 80386er-Klasse, Grafikkarten bis 1280 × 1024 Pixel Auflösung und leistungsfähige Software haben einen professionellen Einsatz von CAD-Layout-Systemen auf PC-Ebene ermöglicht. Das Low-Cost-System Draftsman läßt dem Anwender große Freiheiten bei der Gestaltung seiner Arbeitsumgebung und setzt einige Layout-Erfahrung voraus. **Seite 54**



Digitalisier-Tabletts aller Größen

Mit der wachsenden Verbreitung von CAD-Anwendungen rückt auch die Frage nach geeigneten Eingabegeräten immer mehr in den Vordergrund. Auch die fünf Tabletts des Herstellers Graphtec müssen durch Präzision den höheren Preis im Vergleich zu einer Maus rechtfertigen.

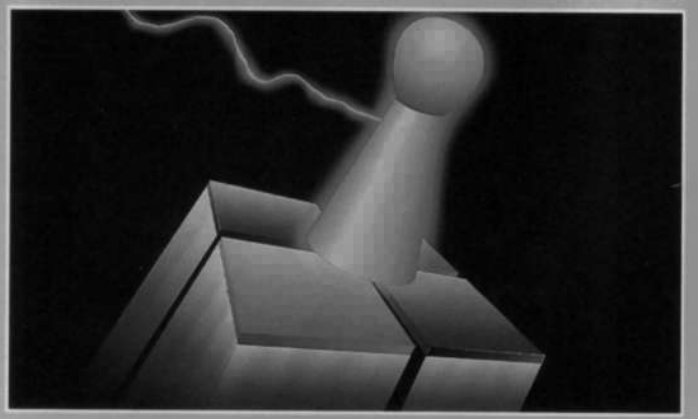
Seite 62



Titan sucht mit Erfolg

Retrievalsysteme sind für die einfache Verwaltung großer Daten- und Textbestände und das problemlose Wiederfinden bestimmter Informationen gedacht. Das System Titan muß beweisen, daß es diesen Anforderungen gewachsen ist. Ob es nicht auch bereits ein Textverarbeitungssystem tut oder ob gar nur ein relationales Datenbanksystem die Aufgaben bewältigen kann, sollen einige grundsätzliche Gedanken zur Stellung von Retrievalsystemen klären.

Seite 65



Modula-2 wird erwachsen

M2SDS, Logitech-Modula, JPI-Modula und die Version Stony Brook werden gegenübergestellt. Damit Sie sich ein Urteil bilden können, werden Handhabung, Compilerfeatures und -fehler, Codequalität, Fließkomma-Arithmetik, Maker, Bibliotheken, Runtime-Systeme und Dokumentation, Utilities, OS/2- und MS-Windows-Unterstützung und der Support verglichen.

Seite 68

INHALT

Potenzieren unter Turbo-Pascal **85**
Eine Lücke im Funktionssatz von Turbo-Pascal wird geschlossen.

Softcopy statt Hardcopy **88**
Ein 24-Nadel-Drucker am Atari ST zeigt, was er kann.

Der Maus-Interrupt 33H **94**
Bei eigenen Programmen bleibt die Maus allzuoft im Loch. Eine Maus-Toolbox hilft ihr auf die Sprünge.

Dynamische Arrays in Turbo-Pascal **102**
Nicht mehr festlegen muß man sich vor dem Start eines Turbo-Pascal-Programms über die Feldgrößen.

Grafik- oder Textmodus **104**
Mit der Logik zum Lightpen gibt die Herculeskarte zu erkennen, in welchem Modus sie ist.

Grafik-Archiv **110**
Abspeicherung von Grafikbildschirmseiten der Herculeskarte.

HARDWARE

Midi-Hardware für Amiga **107**
Mit einer einfach nachzubauenden Schaltung verpaßt jetzt auch der Amiga keinen Einsatz mehr.

DOS-WORKSHOP

Datei-Konverter für dBase **119**
Große Datenmengen und weit und breit kein Instrument zur Auswertung? Ein Programm macht sie für dBase zugänglich.

Mehrere DOS-Kommandos auf einen Streich **131**
Befehlsgewalt unter MS-DOS.

ASSEMBLER-WORKSHOP

Tastaturabfrage leicht gemacht **124**
Der Interrupt 16H zur Ermittlung des Tastencodes reicht für die Gestaltung von Benutzeroberflächen nicht immer aus.

C-WORKSHOP

Aus .C mach .H **132**
Die Compiler/Link-Zeit ist kürzer, wenn man Include- in Objektdateien verwandelt. Die erforderliche Header-Datei wird vom abgedruckten Listing erzeugt.

RUBRIKEN

Editorial **3**
Ray Duncan **6**
Briefe **8**
Info **10**
Bücher **40**
Spruch des Monats **46**
Trend **150**
Markt **154**
Impressum **181**
Vorschau **182**

RAY DUNCAN

Wir in den USA ... ... und OS/2

Es ist seit der offiziellen Freigabe von OS/2 ein Jahr vergangen. Aber Anwender, Händler und Software-Entwickler stehen diesem neuen Betriebssystem von Microsoft immer noch verwirrt und skeptisch gegenüber. Ich glaube, daß es für diese zögernde Haltung in Bezug auf OS/2 im wesentlichen fünf Gründe gibt: Bedenken wegen Hardware-Kompatibilität, eine unerwartete Steigerung der RAM-Preise, Aktivitäten in der Unix-Szene, die Vorab-Version des Presentation Manager zu OS/2 und eine ständige Flut an falschen Informationen und Gerüchten in den Zeitschriften und Computermagazinen. Jeder Faktor sei hier kurz beleuchtet.

Wenn OS/2 laufen soll, gewinnt die komplette Hardware-Kompatibilität zu IBM-ATs und PS/2-Typen größte Bedeutung. In der MS-DOS-Welt konnten viele Hersteller von Nachbauten Hardware-Kompatibilitätsprobleme durch ihr ROM-BIOS überspielen, denn die meisten MS-DOS-Device-Treiber nutzen ROM-BIOS-Routinen für ihre I/O-Operationen und greifen nicht direkt auf die Hardware zu. Ein Betriebssystem wie OS/2, das auf den Schutzmodus des 286 ausgerichtet ist, kann jedoch die I/O-Routinen des ROM-BIOS garnicht benutzen, weil diese nicht reentrant programmiert sind und die Segment-Register nicht adäquat behandeln. OS/2 nutzt das ROM-BIOS während des Bootens und ist dann – einmal hochgefahren – vollständig unabhängig davon. Die Konsequenz: Peripherie-Einheiten, die auf ROM-BIOS-Ebene, aber nicht auf Registerebene kompatibel sind, werden unter OS/2 nicht korrekt arbeiten.

Wie groß ist dieses Problem in der Praxis? Nach meiner Erfahrung läuft die OS/2-Version von IBM fast auf allen AT-Nachbauten prächtig, die von so namhaften Herstellern wie Zenith, Compaq, Dell und ähnlichen stammen. Der entscheidende Faktor in Bezug auf IBM-Kompatibilität scheint der Festplatten-Controller zu sein. Das IBM-OS/2 spielt beispielsweise nicht mit den meisten RLL-Festplatten-Controllern zusammen, die jetzt populär geworden sind. Einen weiteren Knackpunkt (aber vielleicht nicht ganz so lebenswichtig) in Bezug auf Kompatibilität bildet die serielle Schnittstelle. Wer also eine neue Maschine vor dem Kauf überprüfen will, der sollte sich sein OS/2 auf Diskette mitnehmen und das zur Wahl stehende Gerät damit hochfahren. Wenn OS/2 von der Floppy her darauf anläuft und wenn man danach das Inhaltsverzeichnis der Festplatte lesen kann, dann sind die restlichen Probleme in aller Regel klein.

Der seit einem Jahr dramatische Anstieg der RAM-Preise hat mit Sicherheit einen Bremseffekt für die Akzeptanz von OS/2 verursacht. Selbst wenn man die RAM-Preise bezahlen könnte – es ist dann immer noch schwierig wirklich Speicherbausteine geliefert zu bekommen. Vor kurzem wollte ich für meinen Compaq 386 eine 32-Bit-Speichererweiterungskarte kaufen, da wurde mir dargetan, daß ich wenigstens 90 Tage warten müßte, bis Aussicht auf Lieferung bestehe. Weil nun aber IBM für OS/2 1.0 ein Minimum von 2 MByte RAM empfiehlt, für OS/2 1.1 mit Presentation Manger sogar 3 bis 4 MByte, fällt OS/2 aufgrund der Kosten für das damit notwendige RAM mit Sicherheit aus der Kategorie für Spontankäufe heraus. Nur nach reiflicher Überlegung und Überprüfung der persönlichen Liquidität kann man der Anschaffung näher treten.

In den USA ist es jetzt sehr modern, den Speicherbaustein-Engpaß auf Fehler der Politiker zurückzuführen, auf mangelnde Vorausschau der heimischen Halbleiterhersteller und nicht zuletzt auf die japanischen Markteinflüsse. Es ist schwer, auf diesem komplizierten Gebiet das Richtige herauszufinden. Die ganze Misere könnte auch einfach eine Folge der derzeitigen Sonnenaktivität sein, wie manche böse Zungen behaupten. Ich bin zuversichtlich, daß sich die Gesetze von Angebot und Nachfrage – wie immer – durchsetzen werden und sich die Situation in 6 bis 12 Monaten normalisieren wird (vielleicht gibt es dann sogar ein Überangebot von RAM-Chips).

Die jüngsten Wellen, die der Unix-Markt schlägt, sind schwer zu interpretieren. Rückblickend kann man sagen, daß jedesmal, sobald eine neue Generation von PCs erschien, dieselbe alte Geschichte über Unix aufgetischt wurde: Unix würde am Ende doch siegen und den Tischcomputerbereich erobern. Diesmal hätten mich die Unix-Partisanen fast überzeugt: Unix-Händler sprachen schon davon, daß alle existierenden Versionen in einer einzigen Version für den 386er zusammengefaßt würden und daß ausführbare Programmdateien portierbar würden, daß die Diskettenformate standartisiert würden und daß die Benutzeroberfläche in Fenstertechnik gestaltet würde. Unglücklicherweise wurden all diese noblen Absichten ein Opfer der ganz normalen Geschäftspolitik: Jetzt haben wir einen gespaltenen Unix-Markt mit IBM und DEC in dem einen Lager (der Mark führt zu unwahrscheinlichen Allianzen!) und AT&T mit Sun Microsystems im anderen. Ich persönlich habe meine Zweifel, daß dieses neue Kapitel der Unix-Saga anders endet als die Kapitel zuvor. Die Anzahl der Installationen wird weiter wachsen wie zuvor, denn die Anzahl der installierten Maschinen wächst. Aber ehe Unix nicht aufgeräumt wird, einer Schlankheitskur unterzogen wird und modernisiert wird, solange wird es als Betriebssystem für Mikrocomputer keine Bedeutung gewinnen – unabhängig von seiner Mächtigkeit. Auf jeden Fall war OS/2 nie als Konkurrent für Unix gedacht. OS/2 ist ein Einbenutzer-Multitasking-System, das für den Business-Bereich mit großen Ansprüchen an die Interaktionsfähigkeit mit dem Benutzer entworfen wurde. Unix ist ein multiuser- und multitasking-fähiges Sytem für den Einsatz mit anderem Anforderungsprofil.

Der Presentation Manager (PM) ist die Antwort von Microsoft und IBM auf den Macintosh. Es handelt sich um eine graphische Oberfläche, die auf der Idee von der Nachbildung eines Schreibtisch-Szenarios beruht und die Fenster benutzt, um von den nichtssagenden Kommandozeilen wegzukommen und die Lernzeiten für das Bedienen abzukürzen. Der Presentation Manager wird eine Standardkomponente in OS/2 1.1 sein und allen Käufern von Version 1.0 (in den USA) mit einem kostenlosen Update mitgeliefert werden. IBM hat angekündigt, daß OS/2 1.1 im Oktober 1988 ausgeliefert werden wird. Ich würde das gerne glauben, aber wenn man bedenkt, daß die Beta-Version des PM erst ab Mai an die Software-Entwickler ausgegeben wurde, halte ich es für wenig wahrscheinlich, daß OS/2 1.1 mit PM schon 3 bis 6 Monate später die Händler erreichen soll. Vergleicht man das alles mit den Geschehnissen um den Macintosh, so wird es frühestens zwei Jahre nach der ersten offiziellen Auslieferung sein, daß man eine angemessene Auswahl von Anwendungen auf dem Markt sehen kann, die wirklich die Vorteile des Presentation Managers nutzen. Bis dahin werden es sich viele Software-Entwickler überlegen, ob sie ihre Software jetzt gleich an OS/2 1.0 anpassen und die PM-Anpassung später erarbeiten oder ob sie erst einmal abwarten sollen. Diese Entscheidung ist nicht leicht. Die Firma, die hier ihre Produkte früh plaziert, könnte den ganzen Markt besetzen. Es gibt also ein starkes Motiv, jetzt schnell etwas zum Vorzeigen und Verkaufen zu haben.

Auf der anderen Seite ist der Presentation Manger – während er uns das Leben er-

Lesen Sie bitte weiter auf Seite 8.

Druckt Farbgrafiken oder farbige Text/Graphik-Kombinationen im Düsentempo – auch direkt auf Overheadfolie?

Beeindruckt meinen Boß durch Spitzenqualität und ich kann auch noch zwischen 7 Zeichensätzen wählen – einfach per Knöpfchen?

Ist mit unglaublichen 486 Zeichen pro Sekunde unglaublich schnell?
Dann kann es nur der neue FUJITSU DL 5600 sein!

Stimmt!
Ihr Drucker-Füchse!

Jetzt kommt der Super-Sprinter unter den Qualitätsdruckern – der FUJITSU DL 5600. Eine Hochgeschwindigkeitsmaschine, die 5 Drucker in einem vereint: Durch die unglaubliche Rasanz von ca. 200 Zeilen/Minute ersetzt er jeden Banddrucker. Lagerlisten druckt er ruck-zuck. Korrespondenz erledigt er nicht nur im Düsentempo und Top-Qualität, sondern bietet dabei auch eine große Schriftenvielfalt, die durch IC-Karten noch erweitert wird. Kombinationen von Schrift und Bild druckt er ebenso blitzschnell und sauber aus wie professionelle Farbgraphiken.

Dabei ist der FUJITSU DL 5600 höchst einfach zu bedienen. Durch die 16-stellige LCD-Anzeige wissen Sie immer sofort, was läuft. Durch die 90° Papierzuführung dürfen Sie das Wort Papierstau ein für alle Mal vergessen. Per Knopfdruck schalten Sie problemlos von Endlos- auf Einzelblattbetrieb. Und alles läuft so angenehm leise wie auf Samtpfoten ab.

Ein Hochgeschwindigkeitsdrucker also, für den auch Sie sich blitzartig entscheiden sollten.

FUJITSU DL 5600.
Der professionelle Hochgeschwindigkeitsdrucker.
Bringt blitzschnell brillante Ergebnisse!

Bitte senden Sie mir genauere Informationen über

☐ den FUJITSU DL 5600

☐ das gesamte FUJITSU-Drucker-Programm

Name _____

Straße _____

PLZ/Ort _____

Firma _____

mc 9/88 SC

Coupon an: FUJITSU Deutschland GmbH
Rosenheimer Str. 145 · 8000 München 80
Drucker-Telefon-Hotline 089/41 30 11 52

Gewinnen Sie einen FUJITSU-Hochleistungs-Drucker...

und holen Sie ihn sich persönlich in Tokio, Malaga oder München ab!

Teilnahmekarten beim Fachhandel oder bei FUJITSU DEUTSCHLAND GMBH.

Teilnahme völlig unverbindlich!

FUJITSU

Drucker von Japans Computerhersteller Nr. 1

leichtert – für Programmierer eine wirklich komplexe Umgebung, und die meisten MS-DOS-Anwendungen (bis auf die seltenen Windows-Anwendungen wie MS-Excel und Aldus Page Maker) werden für ihn komplett umgeschrieben werden müssen. Keine verrnünftige Firma wird ihre Programme zweimal auf OS/2 umstellen wollen, einmal für die kommandogesteuerte Oberfläche von OS/2 1.1 und dann wieder für die grafisch orientierte Oberfläche von Version 1.1. Was jetzt wirklich nützlich wäre, das wäre eine objektorientierte Programmiersprache für den PM, die den Programmierer automatisch von der Last der vielen Details befreit, die zur Bedienung von PM gehören und die den Entwurf von Anwendungen unter OS/2 1.1 mehr natürlich gestaltet. Aber solch eine Sprache wird auf sich warten lassen.

Zum Schluß noch bemerkt: Ich bin immer noch erschüttert darüber, daß bei uns in den USA so viel totaler Unsinn über OS/2 in der Presse veröffentlicht wird – und das ein Jahr nachdem es überall zu haben ist. Viel von diesem Unsinn muß von Leuten geschrieben sein, die sich auf das Hörensagen verlassen und nicht einmal an einem laufenden OS/2-System gesessen haben können, geschweige denn ein Programm dafür entwickelt haben. Beispielsweise schrieb der Herausgeber von PC World, Richard Landry neulich, daß OS/2 auf jedem Computer unbrauchbar sei, der nicht wenigstens eine 20-MHz-80386-Maschine sei. Weil ich seit eineinhalb Jahren auf einem 8-MHz-AT OS/2-Software entwickelt habe, kann ich diese Feststellung nicht ohne weiteres schlucken. Ein anderes Beispiel: Im July 1988 hat Bruce Tonkin in dem Blatt Computer Language (unter Programmierern ein respektiertes Blatt) einen langen Artikel mit dem Titel „Why OS/2 Must Fail" geschrieben. Darin beklagt er sich, daß OS/2 langsam sei (die meisten Benchmarks zeigen, daß die OS/2-Datei-Verwaltung weit schneller als die MS-DOS-Dateiverwaltung ist). Er klagt weiter, daß OS/2 zu kompliziert sei, um es zu lernen (obwohl die Kommando-Oberfläche eine Obermenge der Oberfläche von MS-DOS ist, und das Interface von PM fast exakt so aussieht wie das von Windows). Außerdem sei es zu schwierig, OS/2-Anwendungen zu schreiben, denn „es ist unmöglich, Multitasking-Programme zu testen" (dieses Faktum dürfte alle Unix-Programmierer überraschen).

Ich will wirklich nicht behaupten, daß der Sieg von OS/2 (oder seines echten 32-Bit-Nachfolgers für den 80386) heute beschlossene Sache ist. Das lange Warten auf OS/2 1.0, die Verschiebungen beim Presentation Manager und die Schwierigkeiten, die es bereitet, wenn man Anwendungen unter PM schreiben muß, das alles eröffnet Gelegenheiten für konkurrierende Betriebssysteme, den Markt zu erobern. Aber der Markt wird erst noch entscheiden. Ihre Entscheidung sollten Sie anhand persönlicher Erfahrungen mit OS/2 fällen und auf verläßliche Quellen stützen, nicht auf die Meinungen von überbezahlten und unterinformierten Magazinschreibern.

BRIEFE

Macintosh und Viren

Zu mc 7/88, Seite 74, „Virusfieber":

Der an sich glänzend geschriebene Artikel enthält leider einen groben Bug, nämlich die Aussage, daß der Macintosh besonders virusgefährdet sei, weil die Disketten beim Einlegen in das Laufwerk nicht schreibgeschützt sein dürfen.

In den einschlägigen ST-Magazinen liest man zwar des öfteren diesen und ähnlichen Unsinn und ich ignoriere das einfach, an mc stellt man aber bekanntlich höhere Ansprüche, daher der Protest. Ursache für diese Enten ist, daß so mancher ST-Besitzer glaubt, etwas vom Macintosh zu verstehen, nur weil auf seiner Maschine der Aladin läuft. Dazu wäre anzumerken:

Es war einmal vor vielen Jahren, da gab es den ersten kleinen Mac mit 64 K ROM und 128 K RAM und etlichen Macken. Die Wunderlampe ist eine Emulation dieser Maschine, und dazu noch eine schlechte. Inzwischen hat sich der Macintosh rasant weiterentwickelt und jetzt eine Perfektion erreicht, die ich dem ST bei aller Liebe auch nicht annähernd bescheinigen kann.

Da ich beide Computer besitze (genau: einen ST 1040, einen Mac SE und Macintosh II), kann ich die Sache beurteilen. Aladin auf einem ST ist ungefähr dasselbe, wie ein Mercedesstern auf einem VW-Kühler. Entsprechend sind dann auch die Aussagen von ST-Aladin-Besitzern zum Thema Macintosh zu werten.

Peter Wollschläger, Hildesheim

Hilfe!

Folgendes Problem wurde von uns in Verbindung des Turbo-Basic-Compilers Version 1.0 und einem IBM PS/2 8550 mit einem EIZO Flexscan 8060S beobachtet:

Die BASIC-Befehle Screen 1: COLOR 9: PRINT "HALLO" erzeugen bei der Ausführung durch IBM BASICA 3.20/3.30 einen dunkelblauen Hintergrund und eine intensivweiße Schrift.

Bei Compilierung mit TURBO BASIC erzeugen sie allerdings einen blaßblauen Hintergrund mit einer grauen Schrift. Auf einem normalen kompatiblen AT mit EGA-Karte sollen die Farben laut Auskunft der Firma Heimsoeth stimmen.

Meine Vermutung ist nun, daß der Compiler auf dem PS/2-System die Farbintensität der VGA-Grafik nicht richtig steuert. Wie läßt sich das Problem beseitigen?

Michael Gaberdiel, Frankfurt

Trick für Stapeldateien

Wer mit einer Stapeldatei ein Verzeichnis komplett löschen will, steht vor dem Problem, auf unerwünschte Nachfragen des DEL-Befehls zu reagieren:

```
DEL C:\TEMP\*.*
Sind Sie sicher (J/N)?
```

Um dem Anwender der Stapeldatei die Beantwortung dieser Frage abzunehmen und den Programm-Ablauf nicht zu stören, muß ein Weg gefunden werden, die Antwort vom Rechner geben zu lassen. MS-DOS erlaubt die Umleitung der Tastatur-Eingabe für ein Programm auf eine Eingabe-Datei. Man schreibt also:

```
DEL C:\TEMP\*.*
<EINGABE.DAT
```

In dieser Eingabe-Datei muß sich das gewünschte „Ja-Wort" befinden: JA<CR>. Diese bekannte Methode braucht aber eine zusätzliche Datei mit sehr geringfügigem Inhalt, die auf einer Festplatte trotzdem 4 KByte belegt. Geschickter ist es, das „JA" in die Stapeldatei einzubetten. Man kann sogar deutsche und englische MS-DOS-Versionen gleichzeitig bedienen:

```
GOTO START
ja
yes
:START
DEL C:\TEMP\*.*
<STAPEL.BAT
```

Mit der hier abgebildeten Stapeldatei ist es möglich, störungsfrei (und unkontrolliert!) das Verzeichnis C.TEMP zu löschen. Für die Richtigkeit der Ausführung ist der Programmierer verantwortlich.

Die hier vorgestellte Lösung erlaubt nur die Bereitstellung eines (allerdings ziemlich wichtigen) Antwortsatzes. Dabei kann nicht verhindert werden, daß das automatische Frage-Antwort-Spiel auf dem Bildschirm dokumentiert wird.

Stefan Demmig 8000 München

Das Haus für Industriecomputersysteme

SE

Σ-X86-19

Der Null "Wait State" Rechner Σ-X86-19*: das Industriecomputersystem.

Speziell für den industriellen Einsatz:

- Robust: stabiles Stahlblech-Chassis
- Zuverlässig: MTBF 3500 h
- Störsicher: spezieller Netzfilter, hochwertige Netzteilmodule
- Color- und Monochrombildschirm
- Wahlweise 3½" oder 5¼" Laufwerke
- SCSI-Interface bis 600 MByte
- LAN-fähig

Das Chassis

- AT-kompatibler Industrierechner in 19"-Einschubtechnik für den Einsatz in rauher Fertigungsumgebung
- Industriegerechter, robuster Hardwareaufbau in 19"/4HE Einschubtechnik
- Front- und Rückplatte aus 3 mm Al beidseitig kunststoffbeschichtet mit geschweißten Befestigungs- und Massebolzen
- Seitenteile und Abdeckblech 1,5 mm Stahlblech gelb chromatiert
- Ausschnitte in der Frontplatte für 2×3½" oder 2×5¼" Laufwerke
- Netzschalter und Lüftungsschlitze
- Schlüsselschalter für Tastaturverriegelung
- Anwendungsgerecht konfigurierbar
- Lieferbar mit kundenspezifischem Firmenzeichen

* Steht für 80386/16 MHz und 80286/10 MHz

SE

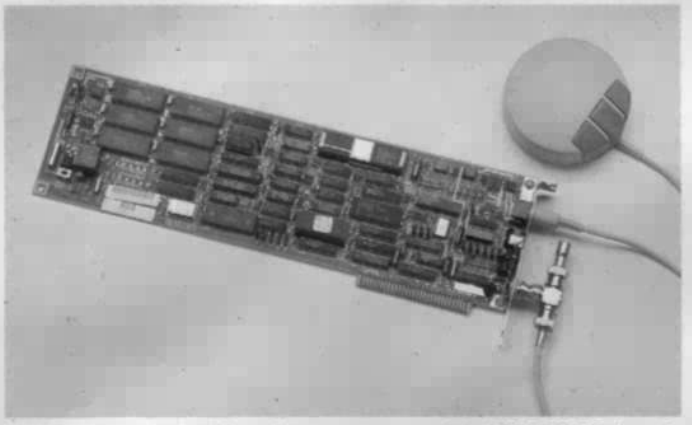
Spezial-Electronic KG

3062 Bückeburg · Zentrale
Postfach 1308, Kreuzbreite 14
Tel. 05722/20306 · Telex 17572210
Teletex 57221O · Telefax 05722/20312O

7090 Ellwangen, Postfach 1320
Dr.-Adolf-Schneider-Straße 11
Telefon 07961/4047 · Telex 17796110
Teletex 7961 10 · Telefax 07961/6030

8000 München 82, Postfach 82609
Karl-Schmid-Straße 8
Telefon 089/4293 33-338 · Telex 17898493
Teletex 898493 · Telefax 089/428137

Ein Beispiel für eine PC-Integration von Digital Equipment



Mit dem Ehternet-/Maus-Adapter lassen sich MS-DOS-PC in ein DECnet/OSI-Netzwerk einbeziehen

VAX/VMS-Dienste für MS-DOS

Digital Equipment kündigte die Version 2.0 von zwei Produkten zur PC-Integration an. Die „VAX/VMS-Dienste für MS-DOS" erlauben die komfortable Integration von MS-DOS-PC in unternehmensweite DECnet/OSI-Netzwerke.

„DECnet-DOS" erlaubt den Anschluß von PC als Endknoten (Peer-to-Peer-Kommunikation) in ein DECnet/OSI-Netzwerk.

Mit den VAX/VMS-Diensten für MS-DOS kann man MicroVAX- oder VAX-Systeme als Datei-, Festplatten- oder Drucker-Server für MS-DOS-PC einsetzen. Informationen und Dienste werden damit netzwerkweit bereitgestellt.

Die Version 2.0 der VAX/VMS-Dienste für MS-DOS erlaubt die Integration von PS/2-Systemen und bestimmten PC-Modellen von Compaq. Die Software unterstützt in der neuen Version NETBIOS-kompatible Applikationen.

Der Fileserver ist gegenüber der bisherigen Version um etwa 30 Prozent schneller. Diese Leistungssteigerung wird durch „multi-threading" erreicht, wobei Anfragen von verschiedenen MS-DOS-Anwendern gleichzeitig am Server behandelt werden.

Die Leistungsfähigkeit der VAX/VMS-Dienste für MS-DOS wurde ferner durch den neuen Local-Area-Disk(LAD)-Server gesteigert. Dieser Diskserver ermöglicht die Ablage von MS-DOS-Dateien auf den Platten eines VAX-Serversystems und kommt dann zum Einsatz, wenn kein Bedarf eines Filesharings mit VMS-Applikationen oder -Benutzern vorhanden ist. Als „virtuelle" Platte ist der so reservierte Plattenbereich des VAX-Serversystems für den PC-Anwender transparent nutzbar und hat eine mit der Festplatte eines IBM PC/AT vergleichbare Geschwindigkeit.

INFO

Die Möglichkeiten zum Systemmanangement und zur Systemsicherheit wurden verbessert. Der neue Local-Area-Disk-Server erlaubt das „remote boot" vom VAX-Server, so daß der Systemverwaltungsaufwand des PC-Anwenders weiter reduziert wird.

DECnet-DOS erlaubt den Anschluß von Industrie-Standard-PC als Endknoten an ein DECnet/OSI-Netzwerk. Die Funktionalität von DECnet-DOS enthält Task-to-Task-Kommunikation, Dateiübertragung, eine Programmentwicklungsumgebung für verteilte netzwerkfähige Applikationen, Terminalemulation, Netzwerkmanagement, Mailfunktionen sowie die Einrichtung von virtuellen Platten und Druckern.

In der neuen Version wurden neben der Benutzeroberfläche von DECnet-DOS Funktionen für Netzwerkmanagement und Terminalemulation wie z.B. die neue Hotkey-Funktion, weiterentwickelt. Auch die Task-to-Task-Kommunikation wurde verbessert. ○

Mit 9 Nadeln 300 Zeichen in der Sekunde

Die Drucker Microline 320 und Microline 321 von Oki gehören in der 9-Nadel-Klasse mit 300 Zeichen/s in DV-Qualität zu den Schnellsten. Neben der hohen Geschwindigkeit bieten sie außerdem eine flexible Papierverarbeitung.

Befindet sich der eingebaute Schubtraktor in der Parkposition, lassen sich Einzelblätter verarbeiten, je nach Bedarf manuell oder mit (optional) automatischem Einzelblatteinzug.

Dank der hohen Durchschlagskraft können mehrere Durchschläge erzeugt werden. Auch ein Papiertransport von unten wird von den Druckern geboten. Ein Zugtraktor ist bei beiden Modellen optional.

Die zwei Versionen des Microline 320/321 verfügen entweder über die Standard-Emulation oder IBM-Proprinter- und Epson FX-Emulationen. Der Microline 320 kostet 1498 DM und der Preis für den Microline 321 beträgt 1898 DM. ○

Zwei neue professionelle und robuste 9-Nadel-Drucker von OKI: Microline 320 und 321

Mit Graylaser lassen sich bis zu 64 Graustufen darstellen

Bilder mit echten Grauwerten

Mit dem Graylaser-System bietet die Stemmer PC-Systeme GmbH aus Pucheim einen Weg, Graustufen auf einem herkömmlichen Laserdrucker mit hoher Qualität auszugeben. Mit je einem Graylaser-Erweiterungsboard für den Rechner (AT) und den Laserdrucker werden bis zu 64 Graustufen pro Punkt ausgegeben. Bei 300 dpi werden bis zu 300 Punkte mit je 64 unterschiedlichen Graustufen pro Inch auf dem Laserdrucker erzeugt. Voraussetzung bei der PC-Hardware ist ein IBM-AT oder ein Komaptibler, ein PFGplus oder FG100 Bild-Erfassungsboard, 2 bis 3 MByte Expanded RAM, ein HP Laserjet+ oder Serie II, zwei Graylaser-Erweiterungsboards. Für die Ausgabe der Bilder über die Graylaser-Zusatzkarte stehen verschiedene Programme zur Verfügung. ○

INFO

Vom Computer zum Dia

Grafiken von PC-CAD-Disketten können bei Kohlhammer in Stuttgart mit Texten gemischt, gestalterisch überarbeitet und in höchster Auflösung auf Farbfilm belichtet werden. Die Grafik-Daten von CAD-Programm wie z.B. AutoCAD und Caddy werden als DXF-Dateien oder im HPGL-Format in ein Grafiksystem eingelesen, das zahlreiche Bearbeitungs- und Gestaltungsfunktionen bietet. So können Farbverläufe eingezogen, Farben geändert oder neu hinzugefügt sowie Texte und Signets eingebunden werden. Die überarbeiteten Bilddaten werden dann konvertiert und an einen rechnergesteuerten Farbfilmrecorder überspielt, der Filmmaterial in den Formaten 35 mm (Kleinbild), 36 mm und 70 mm belichtet. Die Auflösung beträgt dabei derzeit 4000 Linien, die Ausgabe mit 8000 Linien ist geplant.

Gedacht ist dieser neue Service für alle CAD-Anwender, die z. B. ihre Konstruktionen präsentieren wollen, wobei das Spektrum von der Großdia-Präsentation bis zur hochwertigen Reprovorlage reicht. ○

AD Lieferung ab Lager
Computertechnik GmbH

Postfach
2820 Bremen 77
Telefon 04 21/37 13 59
Fax: 04 21/6 36 52 96

Ladenverkauf:
Münchener Straße 58
2800 Bremen 1

11 verschiedene **80286** ATs, z. B.:
10 MHz/0 **waitstates** ab **1869.-**
12 MHz, CPU-Card **999.-**
15 MHz, DSC ab **2499.-**
12 MHz/0 waitstates ab **2299.-**
18,7 MHz 80286-AT auf Anfr.

3× 80386
von
20 MHz ab DM **5299.-**
33 MHz ab DM **7890.-**
80 MB/28 ms nur DM 1450.–
Viele Einzelkomponenten preiswert, z. B. 10 MHz/0 waitstate, AT-Mainboard DM 699.–

AD Turbo **AT**

Wendin-DOS nur DM 260.–
(Multiuser/Multitasking/DOS kompatibel)

dB-Fast dBASE III+ Compiler ... nur DM 290.–
(netzwerkfähig, schnell, kompatibel)

Original Genius Mouse
GM-6 plus DM **99.-**
GM-6 Mouse nur **DM 75.–**
NEC P7 Drucker **DM 1398.–**

Handyscanner, inkl. DPS **DM 520.–**
Handyscanner, superbreit **DM 555.–**
Handyscanner, inkl. Texterkennungssoftware **DM 669.–**

Turbo **XT** ab **929.-**

Einzelkomponenten, AddON, Netze, Drukker, Harddisk, AD/DA-Wandler, IEEE 488 und vieles mehr.

Sofort Katalog anfordern gegen DM 1.– in Briefmarken

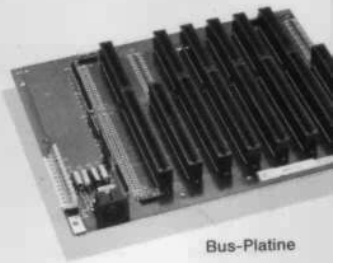
Industrie-Standard i
DSM AT 286 – DSM AT 38

AT-Maschinen von DSM sind modular aufgebaut. Das heißt, Sie selbst bestimmen die Performance Ihres PCs durch die Wahl jener Komponenten, die für Ihre Applikation das beste Preis-/Leistungsverhältnis bieten.

Typisch für das DSM-Konzept ist die Bus-Platine mit insgesamt acht Steckplätzen. Die BUS-Platine enthält keinerlei Logik und ist damit so gut wie 100%ig ausfallsicher. Und sie ermöglicht Ihnen eine applikationsspezifische Konfiguration für Industrie, kommerzielle Anwendung oder Ausbildung sowie das problemlose Upgrading von 16 auf 32 Bit durch einfachen CPU-Wechsel.

Weitere Merkmale:

- 16- und 32-Bit-CPUs
- IBM® PC/AT-kompatibel
- MS-DOS 3.3®
- Solide stabile Bauweise
- Hohe Qualität der Boards
- Extrem zuverlässig
- Deutsche Assemblierung
- Funkentstört nach VDE
- Besonders servicefreundlich
- 1 Jahr Garantie auf DSM-modular-Fertiggeräte



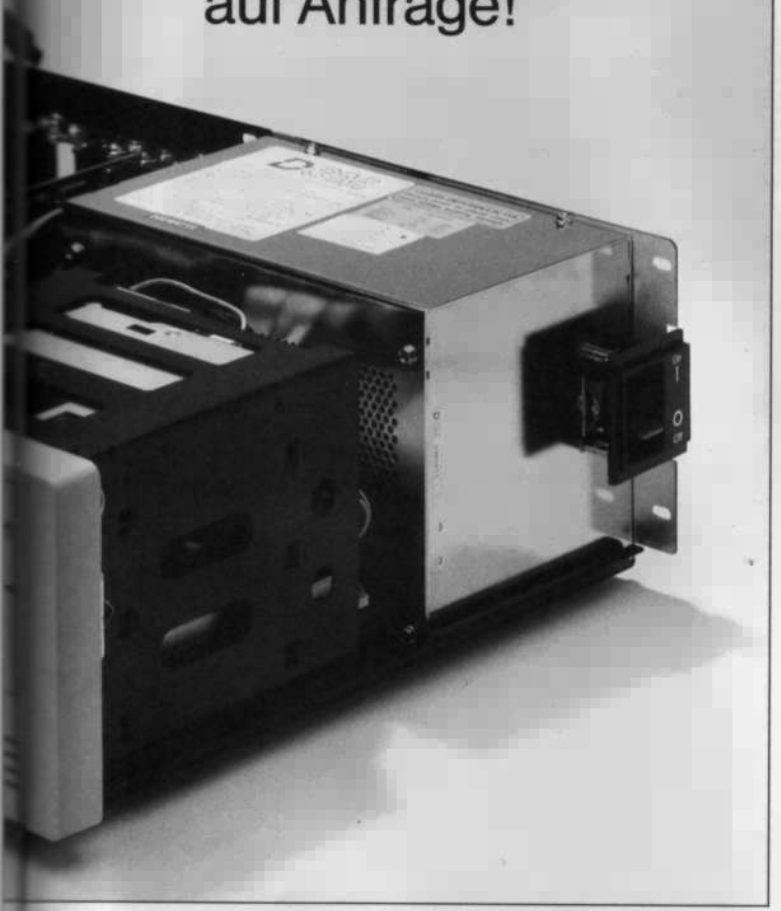
Bus-Platine

m Modular-Konzept

6

NEU!

OEM-Konditionen
auf Anfrage!



Bei den hier aufgeführten Karten handelt es sich um qualitativ hochwertige Module, die auch in herkömmlichen IBM- oder IBM-kompatiblen Systemen einsetzbar sind. Auf alle Komponenten ½ Garantie! Alle Preise inclusive 14 % MwSt.
IBM ist eingetragenes Warenzeichen der IBM Corporation.

1 DSM AT286-modular-Fertiggerät bestehend aus AT-modular-Board V 2.0, 512 KB RAM (100 ns), BUS-Platine, Festplatten-Floppy-Controller, Monochrome-Graphik-Karte, stabiles Gehäuse, 200 Watt Netzteil, MF-2-kompatible Tastatur (102 Tasten), 5¼" 1.2 MB Floppy-Disk-Laufwerk, 20 MB Seagate-Festplatte (65 ms), 14"-Monitor. Mit MS-DOS 3.3®, GW-Basic, und deutschen Handbüchern. Komplett aufgebaut und geprüft (48 Std. Burn-In), 1 Jahr Garantie
Best.-Nr. M0203 **DM 4.198,–**

2 DSM AT286-modular-Kit 2
Wie Pos. 1 als Teile-Kit
Best.-Nr. M0202 **DM 3.798,–**

3 DSM AT286-modular-Kit 1
Wie Pos. 2 ohne Hard-Disk, ohne MS-DOS und Monitor
Best.-Nr. M0201 **DM 2.798,–**

4 AT286-modular-Board V2.0
auf 1 MB RAM ausbaubar, mit 512 KB Speicherbausteinen bestückt, 7 Kanal DMA, 16 Interrupt-Ebenen, 3 programmierbare Timer, 1 Realtime-Clock, CMOS-RAM, bis 64 KB ROM, Keyboard-Controller, 10 MHz, 0-Wait! Neuer CHIPS-TECHNOLOGY-Satz und mit Standard-Speicherbausteinen bestückbar. Award-BIOS. Inkl. Handbuch
Best.-Nr. M0102, 100ns RAMS's **DM 1.298,–**
Best.-Nr. M0100, 120ns RAM's **DM 1.198,–**
Best.-Nr. M0101, ohne RAM's **DM 898,–**

5 AT386-modular-Board
Hochleistungs-CPU-Karte mit 80386 Prozessor (32-Bit) 16 MHz. Bis zu 16 Mbyte-0-Wait-Speicher, Sockel für 80287 und 80387 Coprozessor
Best.-Nr. M0142 **DM 3.798,–**

6 Bus-Platine
mit insgesamt 8 Steckplätzen, davon 6 AT-Slots sowie 2 XT-Slots
Best.-Nr. M0103 **DM 98,–**

7 Festplatten-Floppy-Controller
Western Digital®-Chip-Satz. Verwaltet bis zu 4 Laufwerke, 2 Floppy-Laufwerke (360 KB / 1.2 MB) und 2 Hard-Disk-Laufwerke. Optional 1 zusätzliches Floppy-Laufwerk
Best.-Nr. M0129 **DM 398,–**

8 Monochrom-Grafik-Karte
HERCULES™-kompatibel, mit parallelem Druckeranschluß
Best.-Nr. M0107 **DM 148,–**

9 EGA-Standard-Grafik-Karte
EGA 640x350, 640x200, 320x200, CGA, MDA, 16/64 Farben
Best.-Nr. M0108 **DM 398,–**

10 SUPER-EGA Level 5
800x600, 640x480, 640x350, 640x200, 320x200, 16/64 Farben
Best.-Nr. M0111 **DM 598,–**

11 RAM-Karte
2 MegaByte Extended und Expanded Memory, 10 MHz, (0 KB-RAM) AT-RAM-Erweiterungs-Platine
Best.-Nr. M0112 **DM 398,–**

12 Gehäuse
Stabile Ausführung mit Befestigungsmaterial
Best.-Nr. M0122 **DM 198,–**

13 Netzteil
200 W, kurzschlußfest incl. Netzkabel
Best.-Nr. M0128 **DM 238,–**

14 Tastatur
MF-2-kompatibel, 102 Tasten
Best.-Nr. M0126 Deutsch **DM 298,–**
Best.-Nr. M0127 ASCII **DM 298,–**

15 3½" 1.44 MB Floppy-Disk-Laufwerk
OS/2®-Format (2 MB unform.)
Best.-Nr. M0104 **DM 398,–**

16 3½" 1.2 MB Floppy-Disk-Laufwerk
AT-kompatibel (1.6 MB unform.)
Hersteller: TEAC/NEC
Best.-Nr. M0105 **DM 298,–**

17 5¼" 1.2 MB Floppy-Disk-Laufwerk
AT-kompatibel (1.6 MB unform.)
Hersteller: NEC
Best.-Nr. M0106 **DM 298,–**

18 Festplatten-Laufwerke
a) 20 MB Seagate (65 ms)
Best.-Nr. M0123 **DM 598,–**
b) 40 MB Seagate
Best.-Nr. M0124 **DM 1.150,–**
c) 80 MB Seagate (28 ms)
Best.-Nr. M0125 **DM 2.498,–**

19 Coprozessoren
80287-1 (10 MHz)
Best.-Nr. M0118 **DM 698,–**
80387 (16 MHz)
Best.-Nr. M0119 **DM 1.398,–**

20 Monitore
14" Monitore mit Schwenkfuß
a) grün
Best.-Nr. M0134 **DM 325,–**
b) bernstein
Best.-Nr. M0135 **DM 329,–**
c) **EGA-Farb-Monitor**
Auflösung bis 640x350 Punkte
Best.-Nr. M0136 **DM 998,–**
d) **EIZO-Farb-Monitor**
Auflösung bis 820x620 Punkte für den Betrieb aller EGA-Karten, TTL-Eingang, umschaltbar auf ANALOG-Eingang
Best.-Nr. M0137 **DM 1.798,–**

21 Betriebssystem
MS-DOS 3.3® deutsch mit GW-BASIC und deutschen Handbüchern
Best.-Nr. M0120 **DM 298,–**

22 Zubehör
a) Handbuch AT-modular-Computer im praktischen Kunststoffordner
Best.-Nr. M0130 **DM 39,–**
b) Kabelsatz (2x Floppy und 1x Hard-Disk)
Best.-Nr. M0131 **DM 49,–**

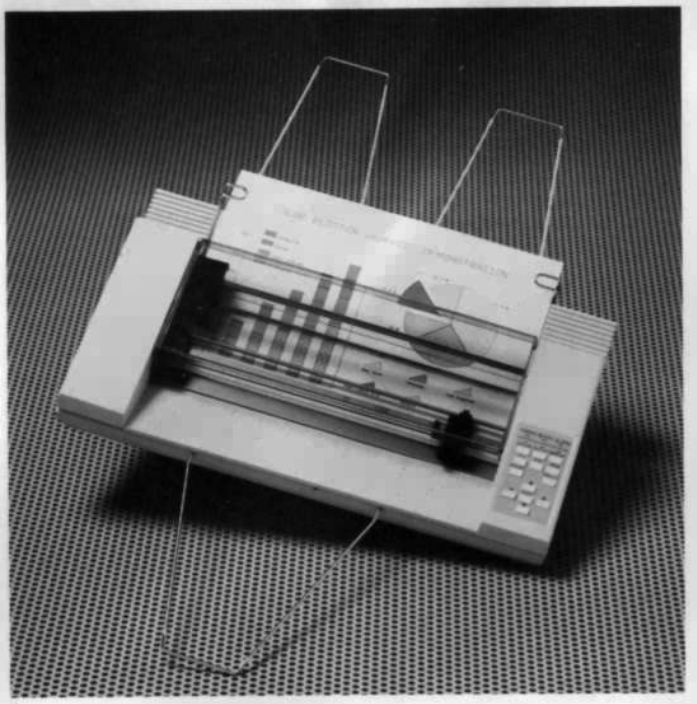
Fordern Sie unsere ausführliche PC-News-Liste an!

Händleranfragen erwünscht.
(Bitte mit Händlernachweis)

DSM Digital Service GmbH

Zentrale: Landwehrstraße 37 · 8000 München 2
Filiale: Schillerstraße 17 · 8000 München 2

DSM **Tel. 089/596916**
Telex 523545 dsm d · Fax 089/598261
Computer Systeme
Ein Unternehmensbereich der *DSM Digital* Service GmbH



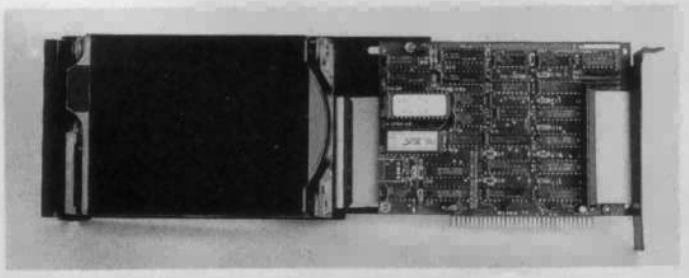
Der Imagegraph eignet sich für den technischen Bereich, aber auch für PC oder Heimcomputer

Plotter von Fujitsu

Der Imagegraph von Fujitsu ist ein 6-Farben-Plotter, der über eine serielle und eine parallele Schnittstelle verfügt. Eine GPIB-Schnittstelle für superschnelle Parallelübertragung ist optional. Auf dem Imagegraph lassen sich Papierformate von der Postkarte bis zu DIN A3 bei einer Geschwindigkeit von 250 mm/sec. verarbeiten. Es stehen Filzstifte, Stifte mit Plastikspitze und Stifte mit Kugelschreiberspitze sowie spezielle Stifte für Overheadfolien zur Verfügung. Die Auflösung liegt bei 0,025 mm. Der Preis für den nur etwa 4,5 kg schweren Plotter beträgt 2275 DM. ○

100-MByte-Festplattenkarte

Mit der Festplattenkarte Wincard 100, die nur 1½ Slots belegt, hat SCSE die Grenze von 100 MByte überschritten. Die Wincard 100 bietet 104 MByte formatierte Speicherkapazität, 28 ms mittlere Zugriffszeit, PC/XT/AT- und 386-Interface mit PC-Bus/SCSI Host Adapter und einem Interleave von 1,1 bei einer Leistungsaufnahme von 11 Watt. Die Wincard 100 kann als zweite oder dritte Festplatte auch bei einem bereits installierten Festplattencontroller zur Systemerweiterung eingesetzt werden. Der Preis für die Wincard 100 beträgt 45 DM pro MByte. ○



Die Wincard 100 wird mit einer menügesteuerten Installations-Software und einer deutschen Einbauanleitung geliefert

INFO

Lernen mit Adcomp

Im September 1988 werden von Adcomp in München folgende Seminare durchgeführt:

- LAN-Informations-Seminar für Fachhändler, 5. und 26. 9. 1988, 19. 9. 88 auch in Düsseldorf
- LAN-Programmierschulung, 5. bis 6. 9. 1988
- LAN-Schulung für Fachhändler, 6. bis 9. 9. 1988
- LAN-Update-Seminar für Fachhändler, 9. 9. 1988
- LAN-Info-Seminar für Anwender, 12. 9. 1988
- LAN-Schulung für Anwender, 13. bis 16. 9. 1988
- LAN-Schulung für Fachhändler, 20. bis 23. 9. 1988 in Düsseldorf, 27. bis 30. 9. 1988 in München
- LAN-Update-Seminar für Anwender, 19. 9. 1988
- Ethernet Fachseminar, 26. 9. 1988
- Token Ring Fachseminar, 27. 9. 1988
- LAN-Marketing-Seminar für Fachhändler, 28. 9. 1988
- LAN-Update-Seminar für Fachhändler, 29. 9. 1988 ○

Computer-Flohmarkt in Düsseldorf

Bereits zum 9. Mal veranstaltet die Volkhochschule Düsseldorf am 1. Oktober 1988 einen Computer-Flohmarkt, bei dem jedermann Computer, Peripherie, Bausteine und alles, was zum Computer gehört, verkaufen und natürlich auch kaufen kann.
Interessenten wenden sich bitte an Herrn Dröse-Star, Tel. 02 11/8 99 34 00. ○

Portables – untragbares Risiko?

Am 20. und 21. September 1988 findet im Messe-Kongreß-Centrum der Messe Düsseldorf der 3. Fachkongreß „Portables/MDE 88" statt. In einer Podiumsdiskussion soll das Thema „Wie sicher sind öffentliche und private Netze, wenn viele, weit verteilte und mobile Computer Zugang zum Netz haben?" angesprochen werden. Unter der Leitung von Dr. Harald B. Karcher äußern sich u.a. Peter Lorenz, Peter von Windau, Horst Görtz (Geschäftsführer der uti-maco-Software GmbH) und Karl-Ludwig Mohr vom Bundeskriminalamt. Zu dem Kongreß, in dessen Mittelpunkt die Einbeziehung portabler Datenerfassungs- und Verarbeitungs-Computer in die unternehmensweite Kommunikations-Struktur steht, werden 500 Fachbesucher erwartet. Ergänzt wird diese Veranstaltung durch eine Hard- und Software-Ausstellung. ○

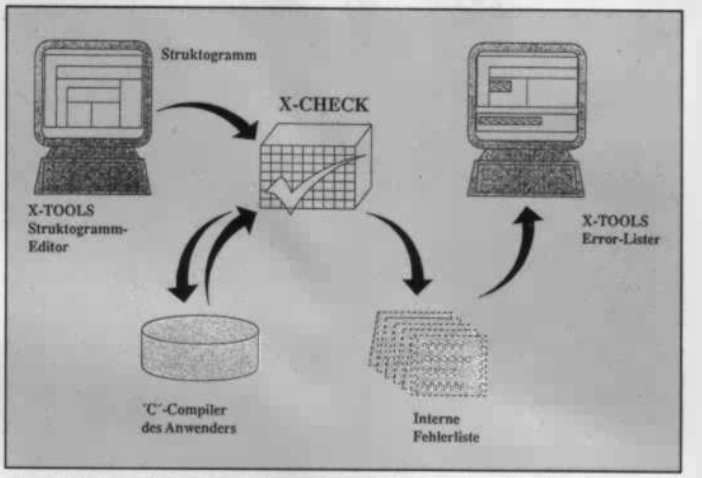
Neue Mitglieder bei OSF

Der im Mai 1988 gegründeten Open Software Foundation (OSF – siehe mc 8/88) sind zwei neue Mitglieder beigetreten: Philips als führendes Unternehmen in der europäischen Computer-Industrie und – als erster Software-Hersteller – die Relational Technology, bekannt u. a. durch das relationale Datenbank-Management-System Ingres (siehe mc 7/88).
Wie bereits berichtet, hat die OSF sich zum Ziel gesetzt, eine offene, übertragbare Software-Umgebung zu schaffen, zu der alle Anbieter und Anwender beitragen können.
Die europäischen Aktivitäten der OSF wird Hennig Oldenburg, bisher bei Nixdorf zuständig für den Bereich CAM Europa, als Direktor leiten. Sitz des OSF-Büros wird Brüssel sein. ○

INFO

PC-Meßgeräte als neue Perspektive

In dem Sonderheft der Siemens-Zeitschrift „Energie und Automation" geht es vor allem um PC-Meßgeräte. In Fachbeiträgen über die neue Generation der Meßtechnik werden verschiedene Anwendungen von PC-Meßgeräten vorgestellt und gleichzeitig dargestellt, wie PC und Meßgeräte zu einem leistungsfähigen System verbunden werden können. Dieses Sonderheft ist sowohl in Deutsch als auch in Englisch erschienen und zu beziehen über den Siemens-Info-Service, Pf. 1 56, 8510 Fürth 2, Tel. 09 11/30 01-2 74. ○



Die Arbeitserleichterung für Software-Entwickler: X-Check von AiD prüft die Syntax von Programmen

Syntax-Analyse automatisch

Die X-Tools-Ergänzungsreihe der AiD GmbH aus Nürnberg bietet als ersten Zusatz einen Syntax-Prüfer namens X-Check. X-Check prüft ein Programm – angepaßt auf den Compiler des Anwenders – hinsichtlich syntaktischer Fehlerfreiheit und Unstimmigkeiten. Die Phasen des wiederholten Compilierens und Editierens werden durch das Zusammenspiel von X-Check mit dem X-Tools-Editor erheblich reduziert und optimiert.

Mit nur einem Befehl wird X-Check gestartet und das Programm geprüft, ohne daß das Entwicklungsprogramm verlassen werden muß. Mit dem erweiterten X-Tools-Error-Lister wird dann Schritt für Schritt durch das Struktogramm gegangen und in jeder fehlerhaften Zeile direkt korrigiert. Damit entfällt das mehrmalige Ausdrucken und die manuelle Fehlersuche.

X-Check arbeitet mit dem Compiler des Anwenders, zu dem er die entsprechenden internen oder externen Fehlertabellen oder „Error-Listings" erzeugt. Die Fehleranzeige erfolgt durch die Bildung von Referenzen zwischen Fehlermeldungen und dem Struktogramm.

Zur Zeit arbeitet X-Check mit Microsoft-C zusammen. Der Anwender kann kleinere Anpassungen an andere Dialekte selbst durchführen. Geplant ist X-Check außerdem für Pascal, PL/M, Fortran und Cobol. Einsetzbar ist der Syntax-Prüfer auf PC/XT und AT unter MS-DOS, auf VAX unter VMS sowie auf IBM 6150 und Apollo unter Unix. ○

VEGA▪VGA zeigt heute...

...256 Farben, aus einer brillanten Palette von 262144

...64 Graustufen, Videoqualität auf PS/2 Monochrom-Monitor

unretuschierte Original-Bildschirmfotos. Copyrights by Z-Soft und IBM.

C&C
Computers and Communications

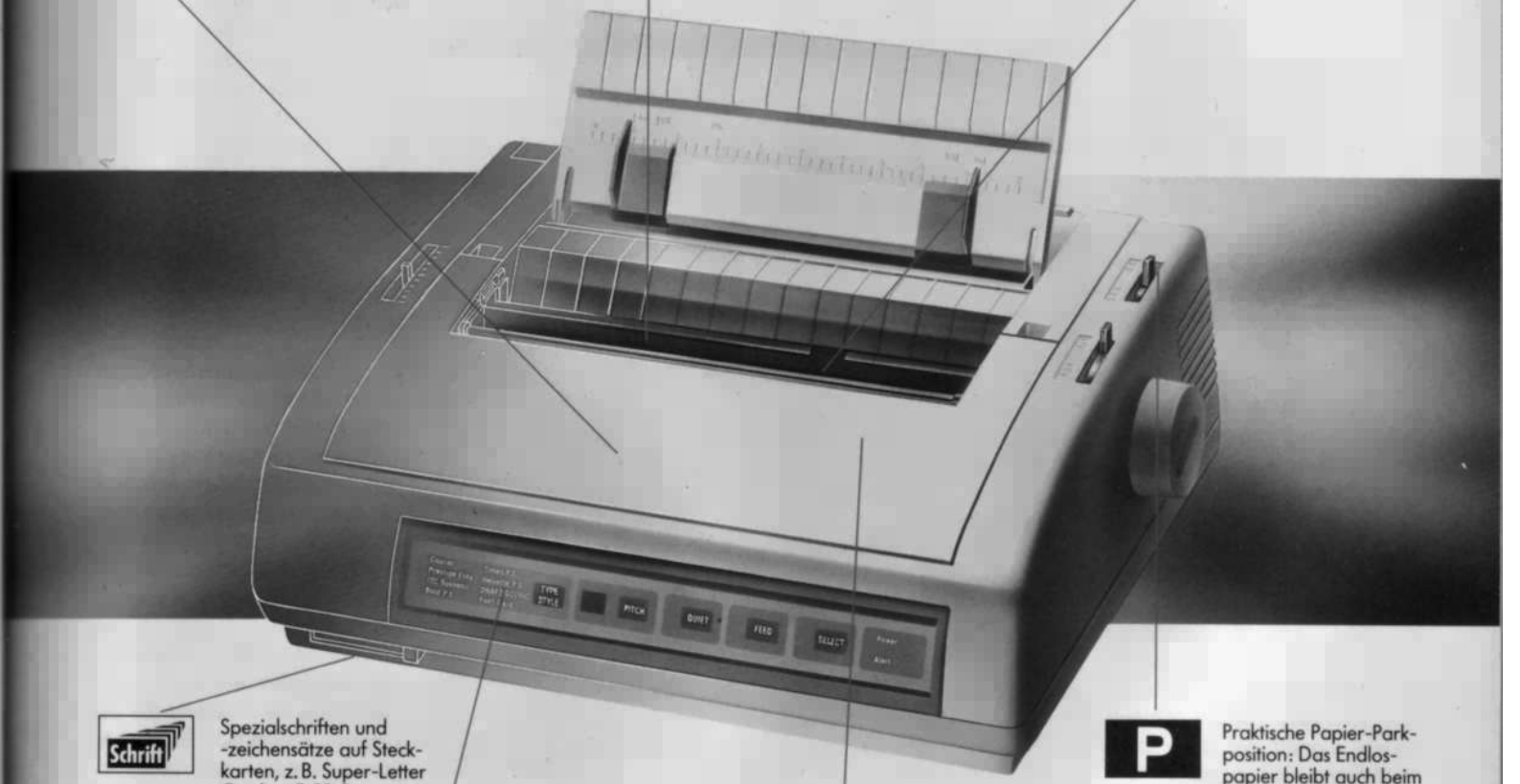
NEU

NEC Pinwriter P6/P7 plus.
Mehr Schriften, mehr Speicher, mehr Möglichkeiten.

80 KB — 80 KByte Pufferspeicher. Bis zu 50 Seiten ausdrucken, ohne den Computer zu blockieren.

COLOR — Farbe einfach nachrüstbar. Carbonbänder für noch besseres Schriftbild.

24 N — 24 Nadeln für hochauflösende Grafik und exzellente Schriftbilder.



Schrift — Spezialschriften und -zeichensätze auf Steckkarten, z. B. Super-Letter Quality, OCR-B, Barcode.

P — Praktische Papier-Parkposition: Das Endlospapier bleibt auch beim Einzelblattdruck im Drucker.

7 Schriften — Sieben einzeln anwählbare Profischriften: Times, Helvette, Prestige Elite, ITC Souvenir, Draft Gothik, Bold PS, Courier.

Ausführliches deutsches Handbuch mit Drucker-Software (incl. PINPLOT). NEC-Hotline-Service für schnelle Informationen.

Die neuen NEC Pinwriter P6/P7 plus sind das Ergebnis konsequenter Weiterentwicklung der bewährten und zuverlässigen Bestseller P6 und P7: Einfache Bedienung, viele Schriftarten und enorm große Speicher. Beide Drucker sind blitzschnell (bis zu 265 Zeichen/Sek.) und garantieren bei einer hohen Auflösung von 360 x 360 dpi feine Grafik- und Schriftdarstellungen, die den Namen „Letter Quality" wirklich verdienen.

NEC

Weitere Informationen erhalten Sie von: NEC Deutschland GmbH, Klausenburger Straße 4, 8000 München 80, Tel: 0 89/9 30 06-0, Telefax: 0 89/93 77 76/8, Telex: 5 218 073 und 5 218 074 nec m

Dieter Strauß

Mikrokanal-Duett

Die neuen PS/2-Systeme kritisch untersucht

Die PS/2-Linie wird von IBM kontinuierlich erweitert. Das Einsteigermodell in die Mikrokanal-Architektur, das Modell 50, wurde verbessert und mit dem Modell 70 eine Lücke in der Produktreihe geschlossen. Beide Computer sind solide aufgebaut. Wie sieht es mit ihrer Rechengeschwindigkeit aus?

Jeder redet über 80386-Rechner, jedoch ist die Mehrzahl der derzeit verkauften PCs der mittleren Preisklasse mit der 80286-CPU bestückt. Dies liegt daran, daß 80286-Systeme immer noch wesentlich billiger als 80386-Systeme sind und sie doch einiges leisten. So ist das IBM-Modell 50 der meistverkaufte PS/2-Rechner, obwohl gerade er in der Öffentlichkeit sehr häufig kritisiert wird. Kritikpunkte sind die geringe Speicherkapazität der Festplatte – nur 20 MByte – ihre langsame Zugriffszeit (85 ms) und die Speicherbausteine mit 120 ns Zugriffszeit. Bei dem Modell 50 Z, das strenggenommen 8550-061 heißt, hat IBM einiges verbessert, um die Kritikpunkte auszuräumen. Die frühere 20-MByte-Festplatte mit ST506-Schnittstelle wurde durch eine 60-MByte-Festplatte mit ESDI-Steuerung (ESDI: enhanced small disk interface) ersetzt. Speicherbausteine mit 85 ns Zugriffszeit ermöglichen den Betrieb des Rechners ohne Wartezyklen, worauf der Buchstabe Z – für zero wait states – in der Typenbezeichnung hinweist.

Bild 1. Das Modell 70 ist ein kompakter Computer

Modell 70 in drei Varianten

Die Bezeichnung Modell 70 führen genaugenommen drei Rechner. Obwohl sie alle mit einem 80386-Prozessor bestückt sind, unterscheiden sie sich gewaltig. Mit 16 MHz wird das von mir getestete Modell 8570-F61 getaktet, mit 20 MHz das Modell 8570-121 und mit 25 MHz das Modell 8570-A21. Nur das Modell 8570-A21 ist mit dem Cache-Controller 82385 von Intel und einem 64 KByte großen Cache ausgestattet. Da der Cache-Controller 82385 nur 32 KByte verwalten kann, wurde von IBM eine Zusatzschaltung zum Verdoppeln des Cache entwickelt. Weitere Unterschiede sind in der jeweiligen Speicherkapazität der Festplatte zu finden. Das Modell 8570-F61 ist mit einer 60 MByte-Festplatte mit 27 ms Zugriffszeit ausgestattet. Die Modelle 8570-121 und 8570-A21 werden mit einer 120 MByte Festplatte mit 23 ms Zugriffszeit geliefert. Bei allen Festplatten werden Dünnfilmköpfe verwendet. Bei Festplattensysteme mit ESDI-Steuerung sind höhere Datenübertragungsraten als bei der konventionellen ST506-Schnittstelle möglich. Bei Dateizugriffen machte das Festplattensystem einen guten Eindruck.

Von außen sind Unterschiede kaum zu bemerken

Das Modell 50 Z und das Modell 70 sehen von außen fast gleich aus. Nur das Modell 70 (*Bild 1*) hat auf beiden Seiten des Gehäuses Schlitze zur Luftzuführung. Im Innenleben und in den Leistungsdaten unterscheiden sich beide Rechner beträchtlich. Angenehm aufgefallen ist mir sofort das relativ geringe Gewicht der Rechner. Verglichen mit den neuen Modellen von Compaq befindet sich wesentlich weniger Blech im System. Laut IBM wird das durch den Mikrokanal ermöglicht, der aufgrund geringerer Kontaktabstände und mehr Masseleitungen weniger Störungen abstrahlt.

Einfache Inbetriebnahme

Die Inbetriebnahme beider Systeme ist einfach. Die mitgelieferte Referenz-Diskette enthält das Setup-Programm und macht den Anwender schrittweise in der Art einer Dia-Show mit dem System vertraut.

Öffnet man die Gehäuse beider Systeme und vergleicht das Innere mit konventionellen Rechnern der AT-Klasse, fällt sofort die Winzigkeit der Speichererweiterungs-Module und die geringen Kontaktabstände der Steckverbinder des Mikrokanals auf. Trotz der geringen Kontaktabstände scheinen die Steckverbinder zuverlässig zu sein. Eine Karte für den Mikrokanal läßt sich sicherer auswechseln als eine Karte für den PC-/AT-Bus, weil Griffkanten angebracht sind. Deshalb sind auch weniger Schäden durch den elektrostatischen Zimmerblitz beim Wechseln von Karten zu befürchten, als in konventionellen Systemen. Das Innere beider Rechner sieht mehr nach High-Tech aus, als bei Systemen mit dem PC-/AT-Bus.

Auf beiden Systemen liefen die Textverarbeitungsprogramme Word und Euroscript und das Grafikprogramm Harvard Graphics ohne Probleme. Die Ergebnisse der mc-Benchmark-Programme sind in *Bild 2* und *Bild 3* zu sehen. Das Ergebnis des Testprogramms PRIM1, das sowohl die CPU-Geschwindigkeit als auch die Video-Ausgabe

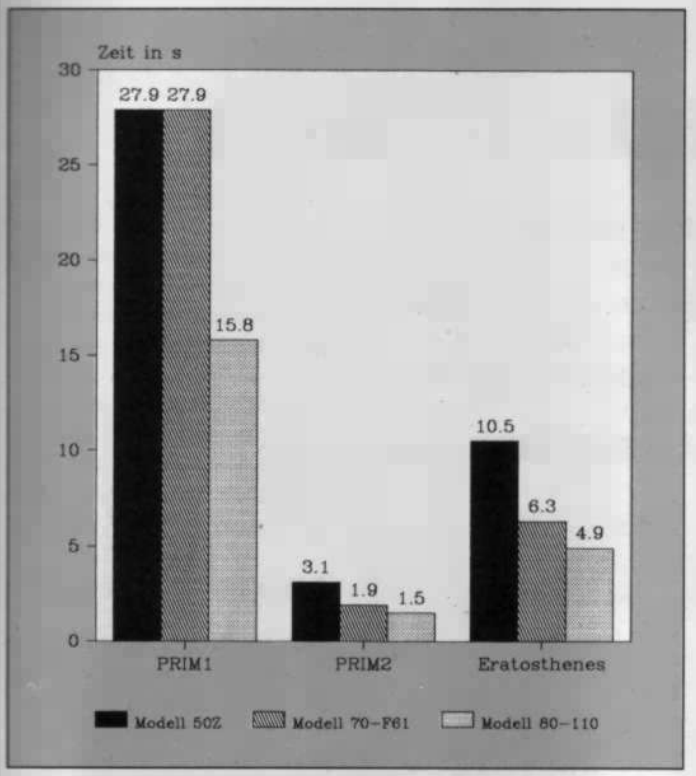


Bild 2. Mehrere Mikrokanal-Systeme im Vergleich

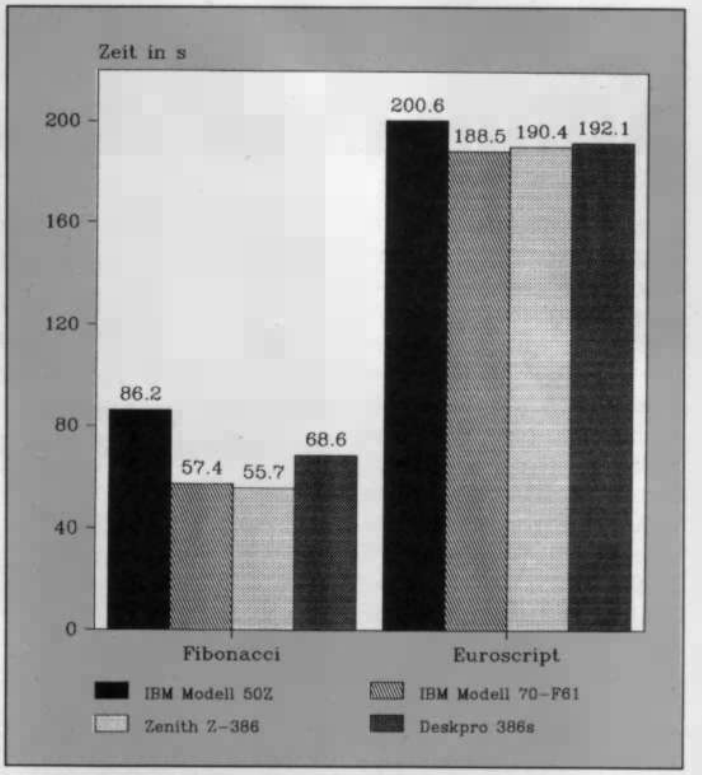


Bild 3. Vergleich der PS/2-Systeme mit anderen Rechnern

im Textmodus mißt, ist etwas enttäuschend. Sogar im normalen Betrieb fällt – verglichen mit anderen Rechnern dieser Klasse – die relativ langsame Ausgabe im Textmodus auf. Alle anderen Ergebnisse liegen im üblichen Rahmen. Beim Fibonacci-Testprogramm, das keinen speziellen 32-Bit-Code enthält, liegt das Modell 70 sogar deutlich vor dem Deskpro 386s von Compaq, obwohl beide Rechner mit der 80386-CPU bestückt sind und mit der gleichen Taktfrequenz arbeiten. In diesen Test geht die Schnelligkeit der verwendeten Speicher und die Speicherorganisation stark in das Ergebnis ein. Auch beim Suche- und Ersetze-Test mit dem Textverarbeitungsprogramm Euroscript schneidet das Modell 70 gut ab. Bei dem Euroscript-Praxistest zeigt sich, daß das wesentlich langsamer getaktete Modell 50 bei diesem Programm nur ungefähr 5 Prozent langsamer als wesentlich schnellere Rechner ist. Das Ergebnis demonstriert, daß ein Programm unter Umständen auch den schnellsten Rechner ausbremsen kann.

Bild 4 gibt das Leistungsprofil der VGA in Verbindung mit der jeweiligen Rechnerumgebung wieder. Im Textmodus mit Scrollen ist der Unterschied minimal.

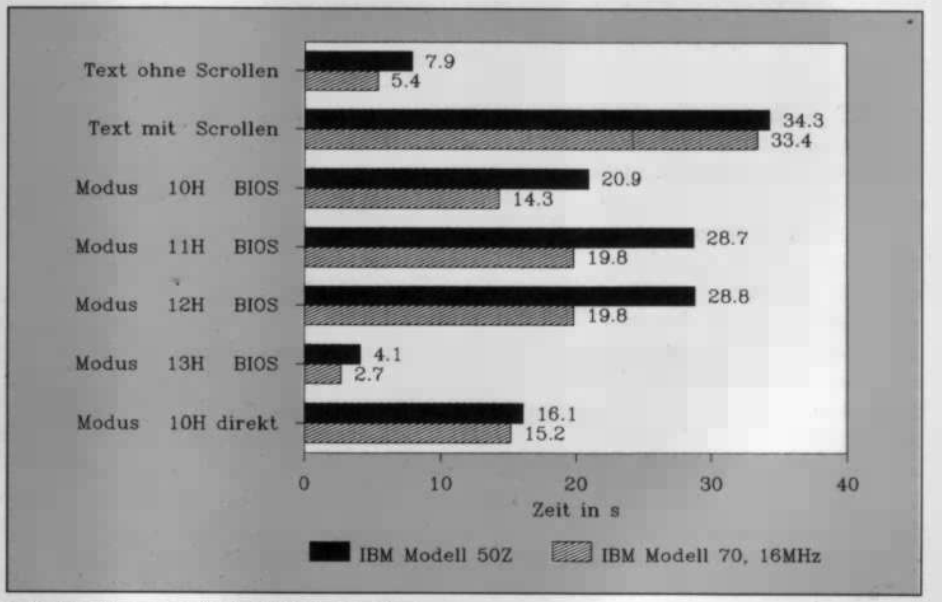


Bild 4. Leistungsprofil der VGA beider Systeme

Konservative Entwurfstechnik

Aufgrund der Testergebnisse und des Vergleichs mit anderen Herstellern, liegt der Schluß nahe, daß IBM nicht das Hauptgewicht darauf legt, der Hersteller mit den schnellsten PCs zu sein. Auch bei Großcomputern ist IBM nicht der Hersteller mit dem schnellsten Modell. Viel mehr Wert scheint IBM auf eine große Produktpalette zu legen, in der es fast für jede Anforderung einen entsprechenden Computer gibt. Nur so ist die Vielzahl der PS/2-Modelle zu erklären. Wie unterscheidet sich z. B. das Modell 70 von dem bereits von mc getesteten Modell 80? Im wesentlichen nur in der Anzahl der Erweiterungssteckplätze und der Größe der Festplattenkapazität. Das Modell 80 ist deswegen auch als Turm-Gehäuse ausgeführt. Daß die PS/2-Modelle – obwohl mit High-Tech-Innenleben ausgestattet – nicht die schnellstmöglichen Computer im PC-Bereich sind, muß offenbar Absicht sein. Konservativer Entwurf geht bei IBM vor Schnelligkeit.

Am Beispiel des Modell 50 Z wird das besonders deutlich. Viele PC-Hersteller bieten bereits 80286-Rechner an, die mit 10 MHz und 120 ns-RAMs ohne Wartezyklen fahren. Möglich ist das durch die „Early

Bitte lesen Sie weiter auf Seite 47

Dieter Strauß

Kompakter AT aus Taiwan

Während früher Taiwan-PCs oft nur preiswerte Nachbauten des Originals waren, bieten sie inzwischen auch von der technischen Seite her gesehen einiges. Einen der neuen Taiwan-Computer – den MPC 2000VE von Mitac – haben wir uns in der Redaktion genauer angesehen.

Bereits bei unserem vorläufigen Test (mc berichtete darüber in der letzten Ausgabe) hinterließ der MPC 2000VE (*Bild 1*) den Eindruck eines sehr schnellen Computers der AT-Klasse. Er ist mit einer 80286-CPU bestückt, die mit 16 MHz getaktet wird.

Angesichts der zunehmenden Verbreitung von 3½-Zoll-Disketten ist es gut, daß der MPC 2000VE Disketten im PS/2-Format (1,44 MByte) und im 720-KByte-Format lesen und beschreiben kann. Ein zusätzliches 5¼-Zoll-Laufwerk macht den MPC 2000VE zu einem idealen PC für Anwender, die Dateien zwischen herkömmlichen PCs und Laptops bzw. PS/2-Systemen austauschen müssen.

Auf der Frontseite des Systems befindet sich der Ein-/Ausschalter und die Reset-Taste. Beide sind so angebracht, daß man sie nicht versehentlich mit der Tastatur betätigen kann, wenn man diese unmittelbar ans Gehäuse stellt.

Außer der üblichen Betriebs- und Festplattenanzeige befindet sich eine Turbo-LED genannte Anzeige auf der Frontseite. Sie leuchtet auf, wenn das System mit 16 MHz getaktet wird.

Die Tastatur macht einen guten Eindruck. Ihr Druckpunkt ist jedoch nach meinem Geschmack zu schwach ausgeprägt. Etwas nervend sind die akustischen Warnsignale, wenn man die Cursor-Tasten bei ausgeschalteter Num-Lock-Taste betätigt. Das BIOS von Phoenix sorgt für größtmögliche Kompatibilität. Vorteilhaft für den Anwender ist das im BIOS integrierte Setup-Programm, das man während des Bootens mit der Esc-Taste aufrufen kann.

Insgesamt passen vier Massenspeicher in den MPC 2000VE. Drei Gehäuseausschnitte sind auf der Frontseite des Gehäuses für Festplatten, Floppy bzw. Bandlaufwerk vorgesehen, wobei ein Ausschnitt nur für ein 3½-Laufwerk geeignet ist. Im Innern des Systems befindet sich ein Haltewinkel für eine 3½-Zoll-Festplatte.

Bild 1. Der MPC 2000VE ist ein schneller 80286-Computer

Geliefert wird der MPC 2000VE mit MS-DOS 3.2. Zum Zeitpunkt des Tests waren das MS-DOS-, das GW-Basic und das Benutzerhandbuch noch in Englisch. Wahrscheinlich werden sie kurz nach der Veröffentlichung dieses Beitrags in Deutsch erhältlich sein. Laut Hersteller wird ab September das System mit MS-DOS 3.3 geliefert.

Interessant ist die technische Konzeption des MPC 2000VE. Bei vielen preiswerten PCs ist der Floppy-Controller auf der Systemplatine untergebracht. Dies ist eine gute Lösung, solange der technische Fortschritt den Anwendern nicht ein Floppy-Format beschert, das der eingebaute Controller nicht bedienen kann. Bei diesen Systemen kann man keinen neuen Controller einstecken, da der alte und der neue die gleiche Portadresse belegen. In einem solchen Fall ist entweder eine Bastelei mit dem Lötkolben oder Resignation angesagt. Nicht so beim MPC 2000VE. Seine Schaltung ist so ausgelegt, daß man mit einer einzigen Brücke den auf der Systemplatine befindlichen Floppy-Controller deaktivieren kann.

Offen für die Zukunft

Das Einstecken eines neuen Floppy-Controllers ist kein Problem. Wie sieht es jedoch mit dem Festplatten-System aus? Hier schreitet der technische Fortschritt noch schneller als bei den Disketten voran. Festplattensysteme mit ESDI-Steuerung oder SCSI-Schnittstelle sind im Kommen. So setzt IBM in den PS/2-Modellen zunehmend ESDI ein. Zwar reichen derzeit der Mehrzahl der Anwender 20-MByte- und 32-MByte-Festplatten mit ST 506-Schnittstelle aus, doch werden Anwender-Programme laufend umfangreicher. Auch sehen Betriebssysteme wie OS/2 oder Unix große Festplattenkapazitäten gern.

Mitac hat den Festplatten-Controller nicht auf die Systemplatine untergebracht, sondern auf einer konventionellen Steckkarte. So kann man statt dem konventionellen Festplattensystem mit ST506-Schnittstelle eine Festplatte mit ESDI-Steuerung einbauen. Falls auf der Platine des Festplatten-Controllers eines Fremdfabrikates auch der Floppy-Controller untergebracht ist, muß man den auf der Systemplatine befindlichen Floppy-Controller ausschalten.

Für den MPC 2000VE bietet Mitac eine 40-MByte- und eine 100- MByte-Festplatte an. Die einfachste Version wird ohne Festplatte geliefert.

Die VGA ist im Gegensatz zu vielen anderen Geräten nicht auf einer Steckkarte untergebracht, sondern auf der Systemplatine. Auch sie läßt sich mit einer Brücke ausschalten. Das Herz der Grafikschnittstelle

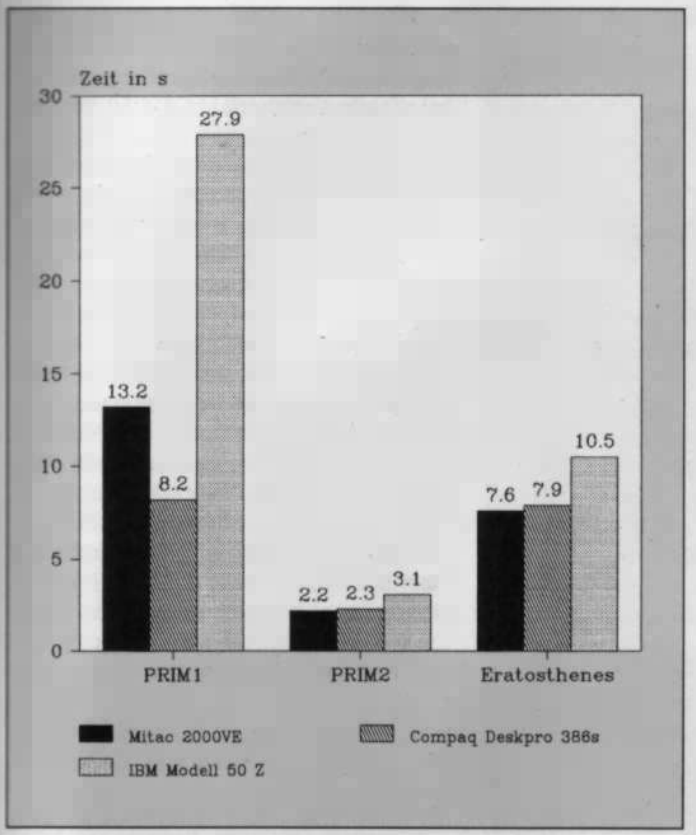


Bild 2. Ergebnisse der Testprogramme PRIM1, PRIM2 und Sieb des Eratosthenes

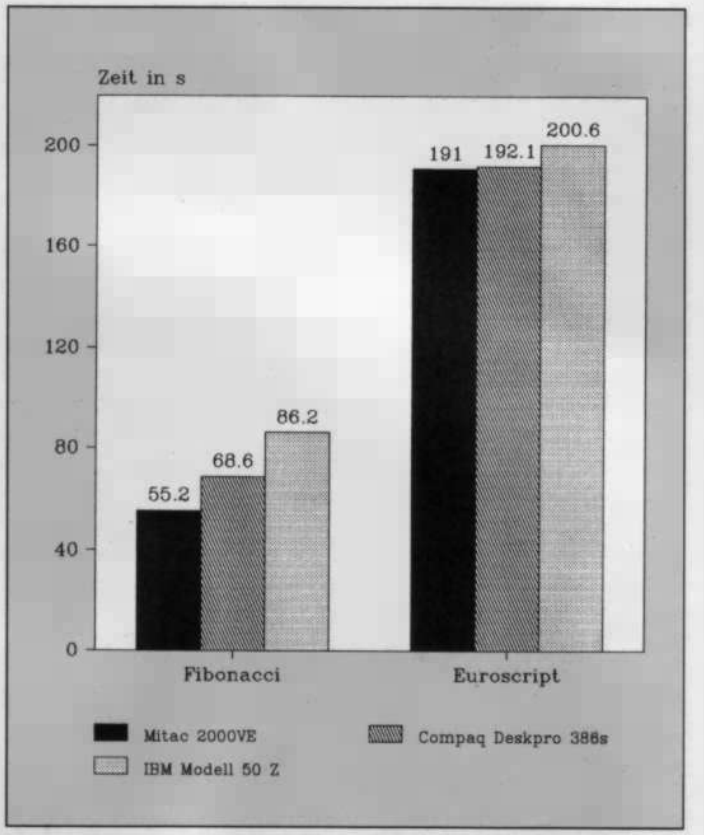


Bild 3. Beim Testprogramm Fibonacci hängt der MPC 2000VE sogar einen 80386-Rechner ab

ist der PVGA1-Chip von Paradise – einer Tochterfirma von Western Digital. Der gleiche Chip wird in vielen Karten nach dem VGA-Standard eingesetzt, so z. B. auch in Compaqs VGA.

Schnelligkeit ist keine Hexerei

Beeindruckend sind die Leistungsdaten des Systems. *Bild 2* und *Bild 3* demonstrieren, daß es der MPC 2000VE bei einigen Testprogrammen locker mit einem 80386-System aufnimmt, das mit der gleichen Taktfrequenz getaktet wird. *Bild 4* gibt einen Eindruck vom Leistungsprofil der verwendeten VGA.

Gemäß dem Motto, daß eine schnelle CPU noch lange kein schnelles System macht, haben sich Entwickler des Systems besonders auf den effektiven Speicherzugriff der CPU konzentriert.

Aus Kostengründen werden für den Hauptspeicher von PCs keine schnellen statischen RAMs, sondern relativ langsame dynamische RAMs verwendet. Bei sehr schnellen Computern der 80386-Klasse gehen daher viele Hersteller dazu über, zwischen der CPU und dem aus dynamischen RAMs aufgebauten Hauptspeicher, einen aus schnellen statischen RAMs aufgebauten Cache einzufügen. Ein spezieller Controller, der 82385, sorgt dafür, daß sich mit

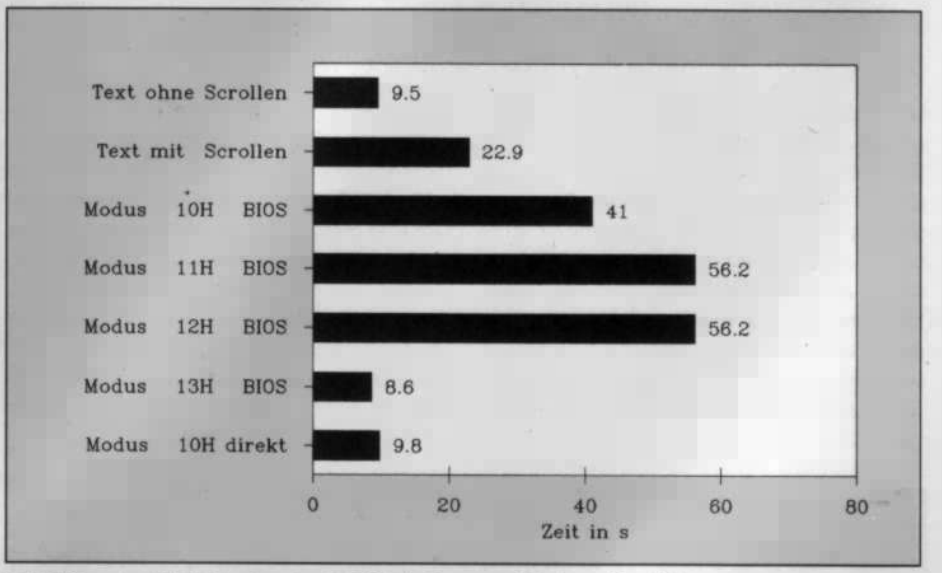


Bild 4. So schnell ist die eingebaute VGA in den verschiedenen Betriebsarten

einer großen Wahrscheinlichkeit die nächsten auszuführenden Befehle und die entsprechenden Daten im Cache befinden. Ein derart hoher Aufwand ist bei einem 80286-PC nicht vertretbar. Dennoch kann man auch ohne Cache einiges tun, um trotz relativ langsamer dynamischer RAMs einen AT fast ohne Wartezyklen zu betreiben. Um dies zu verstehen, muß man etwas in die Theorie der dynamischen RAMs eingehen.

Aus Platz- und damit auch Kostengründen will man die Zahl der Anschlüsse eines dynamischen RAMs möglichst gering halten. Die Zahl der Anschlüsse läßt sich verringern, wenn man die Adresse in zwei Schritten überträgt und im RAM zwischenspeichert. Zwei Strobe-Signale, eines für die Reihenadresse (Row Address Strobe: $\overline{\text{RAS}}$) und eines für die Spaltenadresse (Column Adress Strobe: $\overline{\text{CAS}}$) steuern die

Die technischen Daten:

Typ	MPC 2000VE
Hersteller	Mitac
CPU	80286
Taktfrequenz	8 MHz, 16 MHz
Coprozessor	Fassung für 80287 vorhanden
Speicher	
– Standard-RAM-Kapazität	1 MByte
– Optionen	bis 8 MByte erweiterbar
– max. Speicherkapazität	8 MByte
– Zugriffszeit	100 ns
– Besonderheit	Page-/Interleave-Speicher, EMS-Controller
Diskettenlaufwerk	
– Standard	1,2 MByte, 5¼ Zoll
– Optionen	1,44 MByte, 3½ Zoll
Festplattenlaufwerk	
– Option	3½-Zoll-Festplatte
Steckplätze	
– 8 Bit	1
– 8/16 Bit	5 (davon 4 frei, wenn Festplatte im System)
Schnittstellen	
– Seriell	2 (COM1 u. COM2)
– Parallel	1× Centronics
Tastatur	MF-II-kompatibel
Bildschirmkarte	VGA
– Modi	MDA, HGC, CGA, EGA, VGA
BIOS	Phoenix
Betriebssystem	MS-DOS 3.21
Dienstprogramme	Mitac Advanced Disk Manager
Reset-Schalter	ja
Netzteil-Leistung	145 W
Gehäuseabmessungen (B × H × T)	377 × 158 × 421 mm³
Gewicht	11,5 kg (mit einem Laufwerk)
Sonstiges	Schatten-RAM für System- und VGA-BIOS

Übernahme der jeweiligen „Adreßkomponente" in die internen Zwischenspeicher (Latches) des RAMs. Zwei Speicherzugriffe dürfen nicht sofort aufeinander erfolgen, sondern erst nach der sogenannten Precharge-Zeit [1]. Man unterscheidet zwischen der $\overline{\text{RAS}}$-Precharge- und der $\overline{\text{CAS}}$-Precharge-Zeit. Diese Zeiten versucht man durch eine geschickte Speicherarchitektur zu verstecken. Mit dem Konzept des verschränkten Speichers (interleaved memory) ist das möglich. Dazu teilt man den Speicher in zwei Blöcke auf. Die erste Speicherbank beginnt auf einer geraden Wort-, die andere auf einer ungeraden Wortadresse. Die $\overline{\text{RAS}}$-Signale der beiden Speicherblöcke sind zeitlich so gestaffelt, daß die $\overline{\text{RAS}}$-Precharge-Zeit der einen Speicherbank von dem aktiven $\overline{\text{RAS}}$ der anderen Speicherbank überlappt wird. Diese Technik eliminiert Wartezyklen, solange aufeinanderfolgende Zugriffe jeweils auf die zuletzt nicht aktive Speicherbank erfolgen. Dies ist bei Befehl-Hole-Zyklen i. a. der Fall, nicht jedoch bei Zugriffen auf Daten. Die Trefferquote liegt bei diesem Verfahren um die 50 Prozent. Um die Trefferquote weiter zu erhöhen, hat Mitac das Verfahren des verschränkten Speichers mit Page-Mode-RAMs realisiert. Bei Page-Mode-RAMs ist das $\overline{\text{RAS}}$-Signal aktiv, solange die Zugriffe innerhalb einer Speicherseite erfolgen, die beim MPC 2000VE 1 KByte groß ist. Die $\overline{\text{RAS}}$-Precharge-Zeit entfällt, jedoch muß die $\overline{\text{CAS}}$-Precharge-Zeit weiterhin beachtet werden. Bei der Kombination beider Techniken, dem sogenannten Page-/Interleave-Verfahren braucht man bei Zugriffen innerhalb einer Speicherseite keine Wartezyklen mehr einfügen. Bei einer 16-MHz-80286-CPU kann man das System noch mit RAMs mit einer Zugriffszeit von 100 ns bestücken. Die Trefferrate ist nicht so hoch wie bei einem Cache, jedoch beträgt sie – abhängig von der jeweiligen Anwendung – bis zu 80 Prozent.

Eine weitere Steigerung der Arbeitsgeschwindigkeit wird beim MPC 2000VE dadurch erreicht, daß nach dem Systemstart das ROM-BIOS in den RAM (Shadow-RAM) kopiert wird. Sämtliche BIOS-Aufrufe werden somit vom schnellen RAM statt vom langsamen ROM ausgeführt.

Ein Blick unter die Haube des Systems macht deutlich, warum trotz der beeindruckenden Leistung des Systems die Abmessungen relativ gering sind. Der neueste Chipsatz von Chips & Technologies – NEAT genannt – ermöglicht die Kompaktheit des Systems. Ein Controller nach dem LIM-EMS-Standard (LIM: Lotus, Intel, Microsoft, EMS: Expanded Memory System) ist in einem IC des Chipsatzes, dem 82C212, gleich mit eingebaut. Anwenderprogramme, die den LIM-EMS-Standard zu nutzen wissen, können im MPC 2000VE bereits unter MS-DOS auf bis zu 8 MByte RAM zugreifen. Standardmäßig ist das System allerdings nur mit 1 MByte RAM bestückt.

Der Mitac 200VE ist für Anwender geeignet, die eine sehr schnelle 80286-Maschine benötigen. Mit seinen Disketten-Laufwerken im 5¼-Zoll und 3½-Zoll-Format ist er ideal für Anwender, die mit beiden Diskettenformaten umgehen müssen. Die einfachste Version (ohne Festplatte, ohne VGA-Monitor und 3½-Zoll-Floppy) ist bereits für 5500 DM zu haben.

Spruch des Monats

Der liebe Gott hat den Menschen den Verstand nicht zum Lösen von Differentialgleichungen gegeben, sondern zur Futtersuche.

Spruch des Vaters von W. T. Runge in „Elektronik ist keine Hexerei", Econ 1966.

Literatur

[1] *Ulrich Tietze, Christoph Schenck:* Halbleiter-Schaltungstechnik, Springer-Verlag,Berlin, 1985.

RAS/CAS"-Technik und den unerschütterlichen Glauben an die Halbleiterhersteller, daß deren Speicher meist schneller als aufgedruckt sind. Doch IBM ist eine konservative Firma und legt Wert auf korrektes Design nach dem Verfahren des schlimmst möglichen Falles (worst case). Early RAS/CAS ist durchaus technisch korrekt durchführbar, doch ab einer bestimmten Taktfrequenz sollte bei Speicherbausteinen mit 120 ns Zugriffszeit ein Wartezyklus eingelegt werden. Zwar sind DRAMs meistens schneller als im Datenblatt angegeben – aber manchmal können ihre Parameter genau den kritischen Grenzwerten entsprechen.

Ein System, dessen Konzeption vom schlimmsten Fall ausgeht, muß mit schnelleren und teueren Speichern bestückt werden oder mit einer reduzierten Taktfrequenz arbeiten. Beides verschlechtert auf den ersten Blick das Preis-/Leistungverhältnis eines Computers. Erhöht wird jedoch die Zuverlässigkeit des Systems.

Schwere Entscheidung für den Anwender

Grundsätzlich muß man sich beim Kauf eines PC der gehobenen Preisklasse für oder gegen den Mikrokanal entscheiden. Derzeit ist das entscheidende Kriterium für den Mikrokanal, seine Fähigkeit mehrere Bus-Master im System unter einen Hut zu bringen, noch von geringer Bedeutung. Da jedoch auch das alternative System am Markt – der Macintosh II – mit einem Bus mit ähnlichen Eigenschaften ausgestattet ist, steht außer Frage, daß sich hier in Zukunft einiges tun wird. Auch steigt die Zahl der Erweiterungskarten für den Mikrokanal ständig. Zusätzlich dürfte dieser Markt durch die PS/2-Clones von Tandy und Dell angeheizt werden (mc berichtete darüber in der Ausgabe 7).

Wenn man keine spezielle 80386-Software benötigt und nicht IBMs Unix-Implementierung AIX auf dem System installieren will, reicht ein 80286-System wie das Modell 50 Z aus. Mit einem Preis von 9228 DM liegt das Modell 50 Z allerdings für ein 80286-System bereits am oberen Ende der Skala.

Das Modell 70 ist für Anwender geeignet, die ein schnelles Mikrokanal-System benötigen und nicht die große Anzahl der möglichen Erweiterungen des Modells 80 brauchen. Das getestete Modell 70 mit 16-MHz-80386 ist für 13 121 DM zu haben und ist für gehobene Anwendungen im Desktop-Publishing-Bereich, Ingenieurwesen und als File Server geeignet. Als reine Textverarbeitungsmaschine sind beide Computer viel zu leistungsfähig. □

Technische Daten IBM-Modell 50 Z:

Typ	PS/2-Modell 50 Z (8550-061)
Hersteller	IBM
CPU	80286
Taktfrequenz	10 MHz
Coprozessor	Fassung für 80287 vorhanden
Speicher	
– Standard-RAM-Kapazität	1 MByte
– maximale RAM-Kapazität	16 MByte
– Zugriffszeit	85 ns
– Besonderheit	bis 2 MByte auf Systemplatine erweiterbar
Diskettenlaufwerk	
– Standard	1,44 MByte, 3½ Zoll
– Optionen	Platz für zweites Laufwerk vorhanden
Festplattenlaufwerk	
– Standard	60 MByte (ESDI-Steuerung)
Steckplätze	3 Mikrokanal-Steckplätze (davon einer mit Video-Erweiterung)
Schnittstellen	
– Maus	ja
– Seriell	max. 19 200 Baud
– Parallel	max. 19 200 Baud
Tastatur	MF-II
Bildschirmkarte	
– Modi	MDA, CGA, EGA, VGA
Monitor	Farbmonitor 8513-002
Betriebssystem	PC-DOS 3.3 oder OS/2 (BS/2)
BIOS	Compatibility BIOS für PC-DOS und Advanced BIOS für OS/2
Reset-Schalter	nein
Netzteil-Leistung	94 W
Gehäuseabmessungen (B × H × T)	360 × 140 × 420 mm³
Gewicht	9,55 kg
Sonstiges	Paßwort

Technische Daten IBM-Modell 70:

Typ	Modell 70 (8570-F61)
Hersteller	IBM
CPU	80386
Taktfrequenz	16 MHz
Coprozessor	Fassung für 80387 vorhanden
Speicher	
– Standard-RAM-Kapazität	2 MByte
– maximale RAM-Kapazität	16 MByte
– Zugriffszeit	85 ns
– Besonderheit	bis 6 MByte auf Systemplatine erweiterbar
Diskettenlaufwerk	
– Standard	1,44 MByte, 3½ Zoll
– Optionen	Platz für zweites Laufwerk vorhanden
Festplattenlaufwerk	
– Standard	60 MByte (ESDI-Steuerung)
Steckplätze	Mikrokanal-Architektur
– 16 Bit	1 mit Video-Erweiterung
– 32 Bit	2
Schnittstellen	
– Maus	ja
– Seriell	max. 19 200 Baud
– Parallel	max. 19 200 Baud
Tastatur	MF-II
Bildschirmkarte	
– Modi	MDA, CGA, EGA, VGA
Monitor	Farbmonitor 8513-002
Betriebssystem	PC-DOS 3.3, OS/2 (BS/2) oder AIX
BIOS	Compatibility BIOS für PC-DOS und Advanced BIOS für OS/2
Reset-Schalter	nein
Netzteil-Leistung	132 W
Gehäuseabmessungen (B × H × T)	360 × 140 × 420 mm³
Gewicht	9,5 kg
Sonstiges	Paßwort

Dieter Strauß

Zwei aus einem Stall

Vergleich der neuen 80386-Rechner von Compaq

Gleich zwei neue 80386-Rechner stellte jüngst Compaq vor. Während man den Deskpro 386s als Low-Cost 386-Maschine bezeichnen kann, darf sich der Deskpro 386/25 zu Recht mit der Bezeichnung Workstation schmücken. Worin unterscheiden sich die beiden Computer und wie schnell sind sie?

Kurz nachdem Compaq die beiden neuen Computer vorgestellt hatte, haben wir sie in unserer Redaktion unter die Lupe genommen. Während der Deskpro 386/25 gegenüber dem Deskpro 386/20 in der Hauptsache mit einem um 5 MHz schnelleren 80386-Prozessor bestückt ist und allerlei Feinarbeit an der Bus-Architektur geleistet wurde, ist der Deskpro 386s mit einem neuen Prozessor – dem 80386SX von Intel – bestückt. Eigentlich müßte man „neu" schreiben, denn der 80386SX ist, von der Softwareseite her gesehen, identisch mit dem 80386. In der Hardware unterscheidet er sich beträchtlich. Nur ein 16 Bit breiter Daten- und ein 24-Bit-Adreßbus sind auf die Anschlüsse des 100poligen SMD-Gehäuses herausgeführt.

Viele Kenner der 80386-CPU werden jetzt sicher einwenden, daß man bereits den Standard-80386 in einen 16-Bit-Busmodus umschalten kann. Der Clou beim 80386SX besteht einfach darin, daß die Anzahl der Anschlüsse, d.h. die Gehäuseform, entscheidend in die Kosten des Chips eingeht. In die Kosten eines Rechners geht ebenfalls die Breite des Daten- und Adreßbusses ein. Ein System mit 16-Bit-Datenbus wird immer billiger als eines mit 32-Bit-Datenbus herzustellen sein.

Eine der Aufgaben unserer Testreihe war, festzustellen, ob ein PC mit einer 16-Bit-80386-CPU überhaupt sinnvoll ist, schließlich ist der gute alte AT mit der 80286-CPU bereits mit einem 16-Bit-Datenbus ausgestattet.

Der Haken der 80286-CPU besteht jedoch darin, daß sie den Modus der virtuellen 8086-CPU nicht kennt, und ihre Segmente nur 64 KByte groß sein dürfen, gegenüber 4 GByte beim 80386. Windows 386 und verschiedene Steuerprogramme nutzen den virtuellen 8086-Modus aus, um Anwendungen in mehreren DOS-Fenstern quasi simultan auszuführen.

Mit dem 80386SX ist es möglich, billigere 80386-Systeme anzubieten, die in der Lage sind spezielle 80386-Software auszuführen. Ein Anfang der 80386-Software macht Borland mit Paradox/386. Kenner der Softwarebranche erwarten, daß sehr viele Softwarehäuser, 386-Versionen ihrer Standardprogramme entwickeln, um die Leistungsfähigkeit dieses Prozessors auszunutzen.

Den 80386SX kann man als Nachfolger des 80286-Prozessors bezeichnen, der von Intel nicht mehr weiterentwickelt wird. Da die Software für den 80286 auf dem 80386 läuft, bedeutet dies für den Anwender keinen Nachteil.

Unsere Vermutung ist, daß Compaq mit dem Deskpro 386s eine Welle von 386SX-Computern auslöst. So wie seinerzeit der Deskpro 386 die Ära der 386-Rechner eingeleitet hat, dürfte jetzt ähnliches im Bereich von AT-Computern der mittleren Preisklasse geschehen. Tatsächlich flatterte uns während des Tests eine Meldung auf den Tisch, in der stand, daß NCR eine Einschubkarte mit dem 80386SX im Laufe des Jahres herausbringen wird. Ein entsprechend ausgestatteter PC wurde für das dritte Quartal angekündigt.

Bild 1. Der Deskpro 386s ist ein Einsteigermodell zur 80386-Klasse

Bild 2. Einer der derzeit leistungsfähigsten 80386-Rechner ist der Deskpro 386/25

Blick auf die Innereien

Die Systemeinheit des Deskpro 386s (*Bild 1*) sieht im Vergleich zum „großen Bruder" schicker aus. Dem Deskpro 386/25 (*Bild 2*) sieht man das viele

Blech an. Tatsache ist jedoch, daß sich ein wohlgefälliges Design und eine hohe Anzahl möglicher Erweiterungen gegenseitig ausschließen. Von einem Rechner in der Preisklasse des Deskpro 386/25 wird von den Anwendern gefordert, daß er sich mit zahlreichen Erweiterungen bestücken läßt. Dazu gehören ein Bandlaufwerk (Streamer), eine zweite Festplatte und ein zweites Diskettenlaufwerk. Allein wegen der räumlichen Abmessungen der Peripherie kann man aus einem solchen Computer kein „Schicki-Micki-Modell" machen. Hier ist Robustheit gefragt. Die sieht man bereits, wenn man das Gehäuse öffnet. Die gesamte Festplatten-/Floppy-Einheit ruht auf gewaltigen Stoßdämpfern. Auch weitere Details, die bei billigeren Computern einfach fehlen, fallen auf. So schützt eine einfache Plastikabdeckung, die Einsteckkarten vor unbeabsichtigten Berührungen beim Abnehmen des Gehäusedeckels.

Auch gegen die Abstrahlung von Störstrahlen hat Compaq einiges unternommen. So ist der Cache, der aus statischen RAMs mit einer Zugrifszeit von 25 ns aufgebaut ist, durch einen Metallschirm abgedeckt. Alle elektrisch leitenden Gehäuseteile sind außer der normalen Verschraubung mit einem flexiblen Metallband mit dem Chassis verbunden. Dies dürfte auch notwendig sein, schließlich arbeitet das System mit seinen 25 MHz bereits im oberen Kurzwellenbereich.

Sowohl beim Deskpro 386s als auch bei seinem großen Bruder wird man beim Abnehmen des Gehäusedeckels angenehm überrascht. Auf der Festplatten-/Floppy-Einheit ist ein Aufkleber mit der Lage sämtlicher DIP-Schalter, der Coprozessor-Fassung und spezieller Steckplätze angebracht. Langwierige Suchaktionen auf der Systemplatine bleiben dem Anwender erspart, wenn z. B. im Handbuch auf die DIP-Schalter SW1, SW2 und SW3 verwiesen wird.

Unterschiedliche Tastaturen

Beide Tastaturen sind von hoher Qualität, wobei mir persönlich die Tastatur des Deskpro 386s sogar besser gefällt. Ihr Design ist etwas ansprechender. Am Deskpro 386/25 mißfällt mir, daß sich die Tastaturanschlußbuchse an der Frontseite der Systemeinheit befindet. Deswegen kann man die Tastaur nicht bis zur Systemeinheit schieben, was bei kleinen Schreibtischen mitunter stört.
Einen Schritt in Richtung PS/2-kompatibler Steckverbinder hat Compaq beim Deskpro 386s getan. Als Tastatur- und Mausanschluß werden die wesentlich kleineren

Technische Daten des Deskpro 386s:

Typ	Deskpro 386s
Hersteller	Compaq
CPU	80386SX
Taktfrequenz	6, 8, 16 MHz (wählbar mit Mode-Befehl)
Coprozessor	Fassung für 80387SX vorhanden
Speicher	
– Standard-RAM-Kapazität	1 MByte (16 Bit) auf Systemplatine
– Optionen	1-MByte-Speichererweiterungs-Platine 4-MByte-Speichererweiterungs-Platine Speicher-Module mit 1-MByte- und 4-MByte-Speicher
– max. Speicherkapazität	modular auf 13 MByte ausbaubar
Diskettenlaufwerk	
– Standard	1,2 MByte, 5¼ Zoll (auch als Option)
– Optionen	1,44 MByte, 3½ Zoll oder 360 Kbyte, 5¼ Zoll
Festplattenlaufwerk	
– Standard	Modell 1: ohne Festplatte Modell 20: 3½ Zoll, 20 MByte Modell 40: 3½ Zoll, 40 MByte
– Option	zweite Festplatte möglich
Bandlaufwerk	40- oder 135-MByte (nur als Option)
Steckplätze	
– 8/16 Bit	4
– 16 Bit	1 (für die Speichererweiterungs-Platine geeignet)
Schnittstellen	
– Maus	Mausbuchse wie bei PS/2-Serie
– Seriell	9poliger Anschluß
– Parallel	Centronics
Tastatur	MF-II-kompatibel
Bildschirmkarte	Video-Grafik-Controller
– Modi	CGA, EGA, VGA
Monitor	Monochrom- oder Farbmonitor
Betriebssystem	MS-DOS 3.31
Reset-Schalter	nein
Netzteil-Leistung	Spitze: 175 W, Dauer: 140 W
Gehäuseabmessungen (B × H × T)	375 × 149 × 401 mm³
Gewicht	Modell 1: 12,2 kg Modell 20: 12,7 kg Modell 40: 12,7 kg
Sonstiges	Gehäuseschloß, Paßwort für Tastatur

6poligen Stecker verwendet. Damit man die Anschlüsse unterscheiden kann, ist oberhalb des jeweiligen Anschlusses ein kleines Piktogramm am Gehäuse angebracht. Die asynchrone serielle Schnittstelle ist als 9poliger D-Sub-Stecker ausgeführt. Ihr Anschluß ist auch mit einem Piktogramm gekennzeichnet, dessen Bedeutung jedoch nicht auf den ersten Blick klar wird. Eine herkömmliche Beschriftung wie beim Deskpro 386/25 wäre besser, denn schließlich sind die Anwender von Computern keine Analphabeten.
Trotz des speziellen Mausanschlusses kann man nach wie vor die konventionelle serielle Microsoft-Maus an der 9poligen Schnittstelle betreiben. Man verliert allerdings dann den seriellen Port zur Außenwelt.

Software-kompatibel

Grafik-Programme wie Harvard Graphics und Micrografx Designer und zahlreiche VGA-Demo-Programme bewiesen die Kompatibilität der beiden Rechner zur Original-VGA.
Sogar Windows Version 2 – das im Gegensatz zu Windows 386 nicht die Eigenschaften des 80386-Prozessors ausnutzt – läuft auf dem Deskpro 386s mit affenartiger Geschwindigkeit.
Vielen Anwendern, die in der Computerei noch wenig Erfahrungen haben, fällt es schwer, ein neues System einzurichten. Erfahrene Anwender mit wenig Zeit, geht bei der Einrichtung eines neuen Systems viel Zeit verloren. Um beiden Anwendergruppen das Leben zu erleichtern, hat Compaq der System-Software ein Hilfsprogramm namens Diskinit beigefügt. Diskinit formatiert die Festplatte, richtet eine MS-DOS-Partition ein, kopiert das Betriebssystem auf die Festplatte, richtet Autoexec.Bat und Config.Sys ein und ermahnt den Anwender eine Backup-Kopie des System-Disketten durchzuführen. Mit einem winziges Detail, das jedoch für unerfahrene Anwender sehr wichtig sein kann, sind die System-Disket-

Technische Daten des Deskpro 386/25:

Typ	Deskpro 386/25
Hersteller	Compaq
CPU	80386
Taktfrequenz	6, 8, 16, 25 MHz (wählbar mit Mode-Befehl)
Coprozessor	Fassung für Intel 80387 und Weitek 3167 vorhanden
Speicher	
– Standard-RAM-Kapazität	1 MByte
– maximale RAM-Kapazität	16 MByte (ohne zusätzlichen Steckplatz)
– Besonderheit	Cache-Controller 82385, 32 KByte Cache, Zugriffszeit 25 ns
Diskettenlaufwerk	
– Standard	1,2 MByte, 5¼ Zoll (auch als Option)
– Optionen	1,44 MByte, 3½ Zoll oder 360 Kbyte, 5¼ Zoll
Festplattenlaufwerk	
– Standard	110 MByte
– Option	300 MByte
Bandlaufwerk	40- oder 135-MByte (nur als Option)
Steckplätze	
– 8 Bit	2 (davon ein Steckplatz für Karten mit voller Länge)
– 8/16 Bit	5 (davon 3 frei)
	2 freie Steckplätze
– 32 Bit	1 (belegt von Speichergrundplatine)
– Optionen	Speichermodule für Speichergrundplatine
Schnittstellen	
– Maus	wird an COM2 angeschlossen
– Seriell	9poliger Anschluß (COM1 und COM2)
– Parallel	Centronics
Tastatur	MF-II-kompatibel
Bildschirmkarte	Video-Grafik-Controller
– Modi	CGA, EGA, VGA
Monitor	Monochrom- oder Farbmonitor
Betriebssystem	MS-DOS 3.31
Reset-Schalter	nein
Netzteil-Leistung	Spitze: 220 W, Dauer: 192 W
Gehäuseabmessungen (B × H × T)	503 × 161 × 419 mm³
Gewicht	Modell 110: 20 kg
	Modell 300: 19,1 kg
Sonstiges	Gehäuseschloß, Paßwort

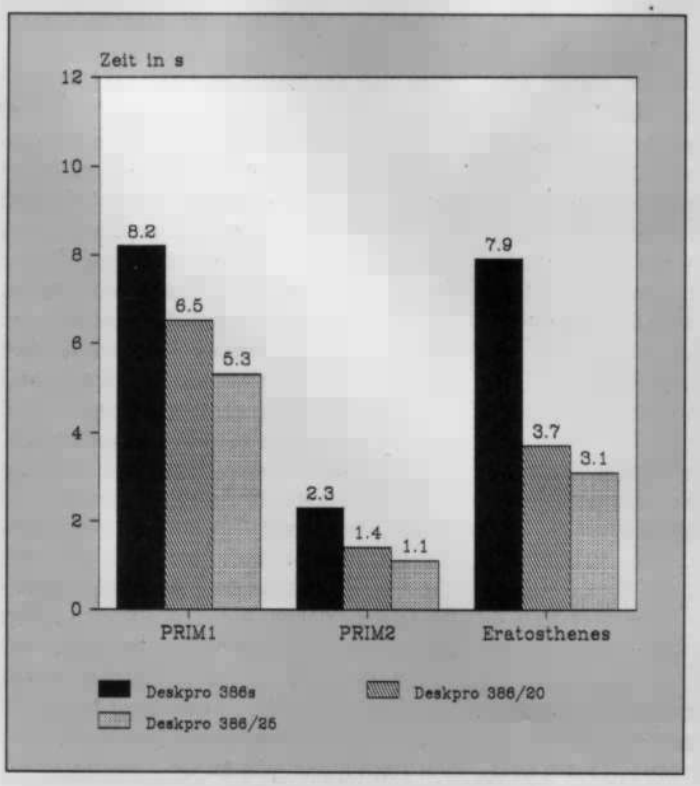


Bild 3. Verschiedene Modelle von Compaq im Vergleich

ten beider Computer ausgestattet: Es sind Disketten, von denen das System nur lesen kann. Ein versehentliches Löschen der Systemdateien ist nicht möglich.

Nette Kleinigkeiten

Jeder PC, von der AT-Klasse aufwärts, speichert einige Konfigurationsdaten in einem batteriegepufferten CMOS-RAM. Eine delikate Sache bei vielen PCs ist die Montage des Akkus, bzw. der Batterie und des Lautsprechers mit doppelseitigem Klebeband. Besonders bei Low-Cost-Modellen kommt es immer wieder vor, daß die Batterie oder der Lautsprecher irgendwo im Gehäuse – nur durch die Anschlußdrähte gehalten – einem Damoklesschwert gleich, am Rande der Systemplatine baumelt. Von dem doppelseitigen Klebeband an dem sie ursprünglich befestigt waren, haben sie sich irgendwann unbemerkt gelöst.
Beim Deskpro 386s wird die Lithium-Batterie durch einen Kabelbinder, beim Deskpro 386/25 durch einen Klettverschluß gehalten. Die Lautsprecher sind mechanisch stabil montiert.
Bei beiden Systemen ist die vom Setup-Programm verlangte Nummer des Festplattentyps am jeweiligen Gehäuseausschnitt aufgeklebt. So kann es kaum passieren, daß man den falschen Festplattentyp mit dem Setup-Programm eintippt und seine Festplatte ruiniert.

Datensicherheit

In der PC-Branche scheint sich allmählich der Gedanke der Datensicherheit breit zu machen. Früher konnten Datendiebe im Nu den Gehäusedeckel öffnen, blitzschnell die Festplatte ausbauen und mit ihr und samt den Daten und Programmen auf Nimmerwiedersehen verschwinden. Daher gehen zunehmend mehr PC-Hersteller dazu über, ihre Systeme mit Gehäuseschlössern auszustatten. Doch Schloß ist nicht gleich Schloß. Und ein Schloß dessen Schlüssel, wie z. B. bei vielen Diskettenboxen sogar noch „kompatibel" ist, d. h. ein Schlüssel öffnet fast alle Schlösser, ist eine Farce. Im Gegensatz zu vielen anderen PCs darf man den Schlüssel zum Deskpro 386s und zum Deskpro 386/25 auch als solchen bezeichnen.
Zusätzliche Sicherheit bietet das Hilfsprogramm KP, mit dem man die Tastatur verriegeln und man erst nach Eingabe des jeweiligen Paßworts weiterarbeiten kann. Selbstverständlich gibt es keine absolute Sicherheit, doch für Anwender, die auf solche Dinge Wert legen, dürfte der Deskpro 386s geeignet sein.

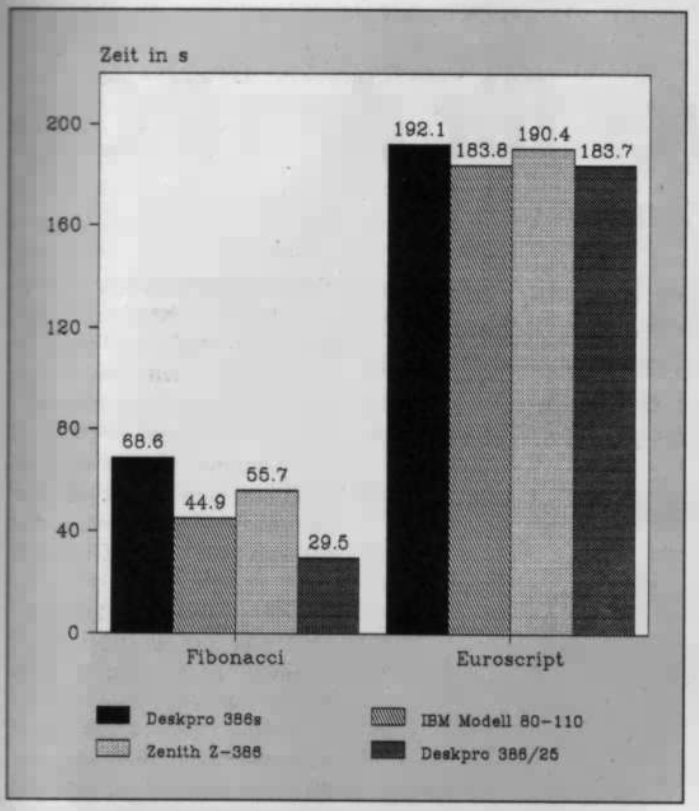


Bild 4. Auch im Vergleich zu anderen Computern schneiden der Deskpro 386s und der Deskpro 386/25 gut ab

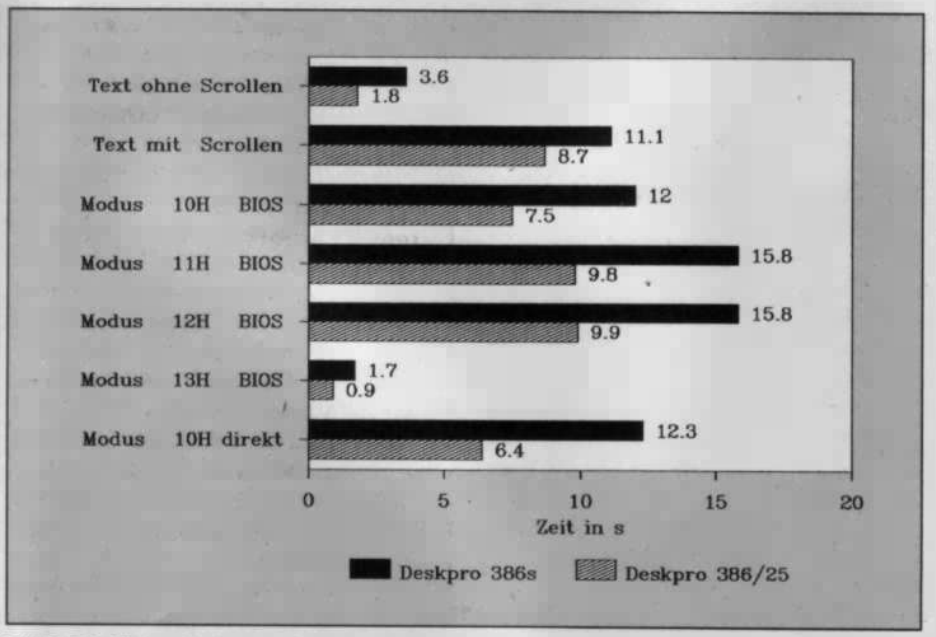


Bild 5. Die Videoschnittstelle ist schnell

Wettrennen

In *Bild 3* sind einige Testergebnisse dargestellt. Die Testprogramme PRIM1 und PRIM2 wurden in mc veröffentlicht [1]. In PRIM1 geht außer der Zugriffszeit auf den Speicher die Arbeitsgeschwindigkeit der jeweiligen CPU und die Ausgabegeschwindigkeit der Video-Schnittstelle im Textmodus ein. PRIM2 läßt nur eine Aussage über die CPU und die Geschwindigkeit des Speicherzugriffs zu. Die Programme Sieb des Eratothenes und Fibonacci wurden erstmals 1983 von der amerikanischen Fachzeitschrift Byte veröffentlicht und dienen seither bei vielen Publikationen als Standard-Benchmark-Programme [2].

Der Euroscript-Test verwendet die Such- und Ersetze-Funktion dieses Textverarbeitungsprogramms. Da Euroscript jeden Ersetzungsvorgang sofort auf dem Bildschirm anzeigt, stellt es hohe Anforderungen an die CPU und die jeweilige Video-Karte. In *Bild 4* vergleichen wir die neuen Compaq-Rechner mit IBM Modell 80-110 und dem Zenith Z-386. Während Fibonacci deutliche Unterschiede zwischen den einzelnen Rechnern zeigt, sind die Unterschiede unter Euroscript so minimal, daß sie ein Anwender nicht bemerken wird. Dies dürfte eher an der Struktur von Euroscript als an den Rechnern liegen.

Die Video-Schnittstelle des Deskpro 386s und des Deskpro 386/25 entspricht dem VGA-Standard. Die 16-Bit-VGA des Deskpro 386/25 ist mit einem Chip von Paradise bestückt. In *Bild 5* legen wir die Ausführungszeiten des mc-VGA-Tests [3] offen. Da beide VGA in unterschiedlichen Computern getestet wurden, sollen die Ergebnisse nicht als isolierter VGA-Test interpretiert werden, sondern als Ergebnis der Kombination der jeweiligen CPU-Umgebung mit der VGA.

Handbücher in Deutsch

Sowohl das MS-DOS-Handbuch, das auch die neuen Befehle von MS-DOS 3.3 erklärt, als auch das GW-Basic- und das System-Handbuch sind in Deutsch. Sämtliche Anschlußbelegungen der Schnittstellen sind dem jeweiligen System-Handbuch zu entnehmen. Ein Detail, das immer noch nicht bei jedem PC-Hersteller üblich ist.

Für wen sind die beiden neuen Computer geeignet? Für Anwender, die einen schnellen und dennoch preiswerten Marken-PC wollen und Wert darauf legen, spezielle 80386-Software ausführen zu können, ist der Deskpro 386s die richtige Wahl. Das Modell 1 des Deskpro 386s kostet ohne Festplatte etwas unter 7970 DM.

Der Deskpro 386/25 ist für Anwendungen geeignet, bei denen es auf höchste Systemleistungen ankommt, wie CAD/CAM. Eine zusätzliche Fassung für den Weitek-Coprozessor erhöht die Flexibilität des Systems. Die Version mit 110 MByte-Festplatte kostet etwas über 20500 DM. Mit dem Deskpro 386/25 kommt der PC in Bereiche, die früher wesentlich teueren Workstations vorbehalten waren.

Literatur

[1] *Strauß, Dieter*: Primzahlen-Sucher. mc 1987, Ausgabe 10, Seite 74.

[2] High-Tech Horsepower. Byte. Juli 1987, Seite 101.

[3] *Sternberg, Günther*: Duell zweier VGA-Clones. mc 1988, Ausgabe 6, Seite 130.

Peter Wollschlaeger

Transputer-Power für den PC

PC mit Transputer-Karte ist schneller als eine VAX

Die Geschwindigkeit, mit der Transputer arbeiten, ist schon beeindruckend. Doch nicht jedermann will deshalb gleich eine neue Maschine kaufen noch sich mit RISC und OCCAM befassen. Neuerdings reicht eine Steckkarte aus, und der PC schlägt sogar Super-Minis.

Ich weiß natürlich, daß mc-Leser auch sehr an der Technik der Transputer interessiert sind. Das haben nicht zuletzt die Reaktionen auf die Serie über die Transputer gezeigt [1]. Aber stellen wir die Technik erst einmal zurück und betrachten die neueste Entwicklung in Sachen Transputer aus der Sicht des Programmierers.

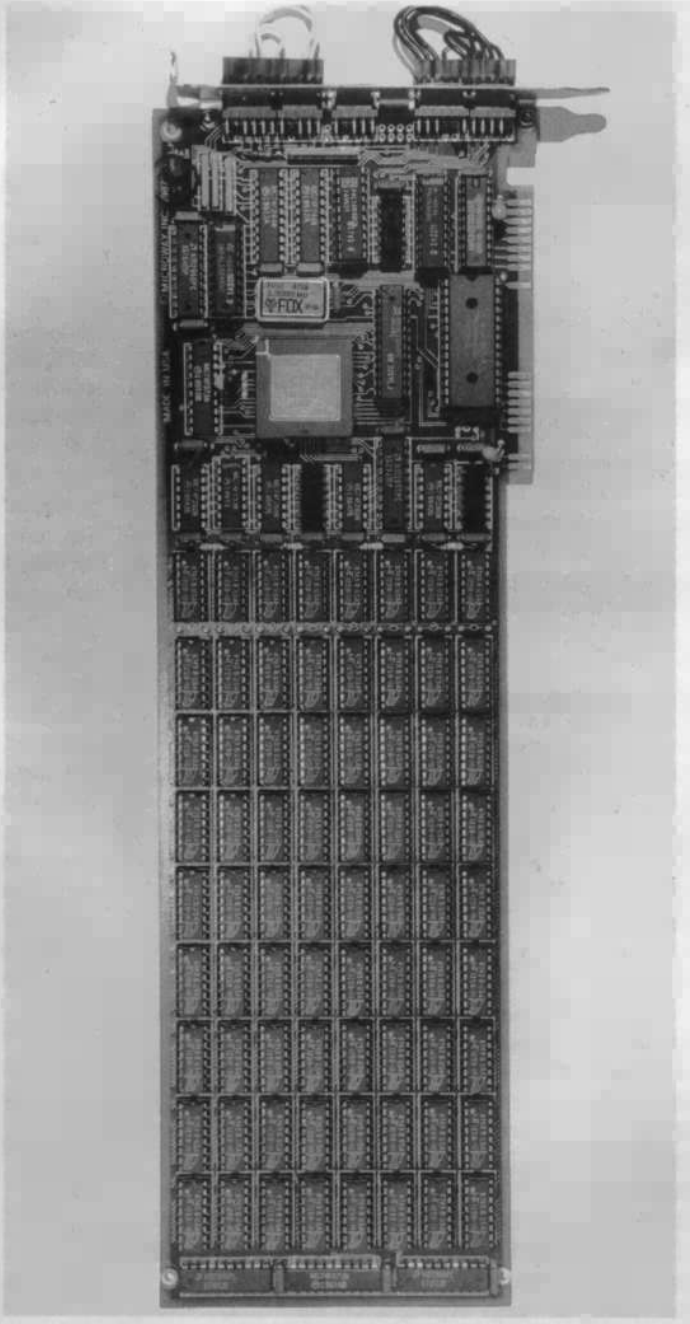
Dieser braucht – sofern er die Leistung seines PC erheblich verbessern möchte – die hier beschriebene Steckkarte und einen Compiler. Wir hatten den sogenannten „Monoputer" von Microway und den dazu passenden C-Compiler von 3L im Test. Einen passenden Fortran-Compiler gibt es auch, aber diese Programmiersprache mag der Tester schon seit der „Computer-Steinzeit" nicht mehr.

Die Installation der Hardware auf einem IBM-AT ist herzlich einfach: Deckel auf, Karte rein, Deckel zu. Anschließend sind noch zwei unverwechselbare Stecker auf die nach außen geführten Kontakte zu stecken. Das sind praktisch Blindstecker, die anstelle möglicher Erweiterungen angeschlossen werden müssen.

Einfache Installation

Ähnlich simpel ist die Installation der Software. In ein neu anzulegendes Directory auf der Festplatte namens „tc1v3" werden die Dateien von zwei Disketten mit „copy a:*.*"übertragen.

Der C-Compiler wurde speziell für einen Transputer geschrieben. Er erkennt selbst, wie viele CPU im System vorhanden sind und richtet sich bei der Übersetzung von Source-Dateien in ablauffähige Programme danach. Anstelle des hier getesteten Monoputers ist auch eine Steckkarte mit vier Tansputern verfügbar. Damit wird zwar das System nicht viermal schneller, doch immerhin etwa um den Faktor 3 bzw. um den Faktor 300 und mehr, wenn man sich auf den AT bezieht.



Der Compiler ist das richtige Tool, um den Supercomputer nutzen zu können. Schauen wir uns also zuerst das 3L-Produkt an.

Geliefert werden ein Compiler, ein Linker und ein Lader. Dazu kommen noch Library- und Header-Dateien. Ein Editor fehlt, aber jeder ASCII-Editor tuts auch. Aufgerufen wird der Compiler schlicht mit „t8c datei.c". Zusätzlich können noch die folgenden Optionen eingeschaltet werden:

/C: Nur Syntax-Check.
/H: Generiert Standard-Hex-ASCII.
/I: Ausgabe der Compiler-Version auf StdOut.
/L: Listfile mit Zeilennummern.
/M: Makroexpansion im Listfile.
/PCn: n Bytes für Calls externer Routinen (mit n=8 16 MByte).
/S: Float-Double-Konvertierung entfällt, wenn Operanden Float.
/T8A: Umgeht einen Bug in der Fließkommarechnung des T800A.
/V: Zusätzliche Messages beim Compiler-Lauf.

Während der Compiler arbeitet, entstehen drei temporäre Dateien auf der Festplatte, die anschließend wieder gelöscht werden. Der Punkt ist insofern erwähnenswert, als daß die Geschwindigkeit des Compilers im wesentlichen von der Zugriffszeit der Festplatte abhängt.

Der Linker kann mit einer Dateiliste oder mit einer Input-Datei (Text) aufgerufen werden. Ich habe eine kleine Batch-Da-

tei vorgezogen, womit sich der Aufruf auf „t8clink Datei" reduzierte. Der Linker kennt nur eine Option, nämlich „/l" zum Erzeugen von Libraries.
Bis hierher spielte sich noch alles in der PC-Welt ab. Jetzt geht es darum, das Programm in den Transputer zu laden und dort zu starten. Dafür gibt es natürlich auch ein Programm. Es heißt „afserver", und wird mit
afserver -:b file.b4 parameter
aufgerufen. Das war mir zu umständlich und wurde deshalb mit einer Batch-Datei auf „run datei" reduziert.
Die ersten kleinen Programme liefen – nach dieser Art compiliert – auch einwandfrei, doch beim ersten „dicken Brocken" kam es zum Crash. Ein Blick in das Handbuch brachte auch schnell die Erklärung dafür. Nach der oben beschriebenen Art compilierte Programme müssen sich mit einem Stack von weniger als 2 KByte (512 Worten) bescheiden. Der Transputer setzt dann sein On-Chip-RAM ein, wodurch sich das Tempo steigert. Nun kommt es darauf an, wieviele Auto-Variable das Programm und die Library-Funktionen belegen (letztere bis zu 130) bzw. in wie vielen Ebenen rekursive Funktionen berechnet werden.
Es gibt nur eine Abhilfe, nämlich den Afserver mit der Option „- :o 1" aufzurufen. Dadurch wird für den Stack generell der externe Speicher (in der Testkonfiguration 2 MByte) benutzt.
Hier würde ich mir wünschen, daß man per Compilerdirektive den Stack umschaltem kann, was zugegebenermaßen schwieriger ist, als es sich anhört.

C etwas mager

Der 3L-Compiler basiert auf dem 3L-C von Lattice Logic. Es handelt sich hier primär um ein C nach dem Kernighan & Ritchie-Standard mit geringfügigen Abweichungen. Die Runtime-Library von 3L bietet einige Extras. Fast komplett sind die Mathematik-Routinen, „ctype", „stdio", und „string".
Sehr vielseitig sind die DOS-Funktionen inklusive Zugriff auf das BDOS und das Enviroment, ausgesprochen mager hingegen sieht es in „time.h" aus, wo sich nur „clock()" findet.
Besonders zu erwähnen ist hier natürlich „Channel I/O", was das Senden und Empfangen von Nachrichten über OCCAM-Kanäle erlaubt. Damit können C-Programme mit parallel laufenden OCCAM- Prozessen kommunizieren.
Alles in allem ist das C-Paket etwas mager, wenn man es mit MS-C oder Turbo-C vergleicht, die dann doch so rund 200 Funktionen in ihren Bibliotheken mehr bieten. Andererseits war zu beobachten, daß es

```
#include <math.h>
#include <time.h>
#define NITER 1000000

main()
{
  long i;
  double x,y;

  printf("%s\n","Start");
  x = 0.0;
  y = 4711.471147114711;

  for(i=0; i< NITER; i++)
    x += y;

  for(i=0; i< NITER; i++)
    x -= y;

  printf("\nFehler = %1.10f",x);
}
```

Bild 1. Dieses Programm führt der Monoputer 200mal schneller als ein 6,7-MHz-AT aus

```
#include <math.h>

main()
{
  long i, iter;
  double x,y;

  printf("Wieviel Iterationen? ");
  scanf("%ld",&iter);
  x = 0;

  for(i=0; i< iter; i++)
  {
    x += (double) 1 / iter;
    y = sin(x);
  }

  printf("\n%1.10f",y);
}
```

Bild 2. Mathematische Funktionen und Fließkommazahlen sind des Transputers Stärke

ziemlich einfach war, MS-C-Programme soweit abzurüsten oder umzuschreiben, daß sie unter 3L-C liefen.
Auch wenn man jetzt ein paar C-Zeilen mehr schreiben oder sich die Bibliotheken erst zusammenstellen muß, wird man doch für alle Mühen reichlich entschädigt. Die Programme laufen mit einer sagenhaften Geschwindigkeit ab.
Zuerst die schlechte Nachricht: für reine Integer-Arithmetik wie sie der Sieve-Test anwendet, lohnt sich die Anschaffung des Transputer-Boards kaum. In diesem Fall ist der Monoputer nur um den Faktor 3 schneller als ein AT-02. Und das resultiert ganz einfach aus der höheren Taktfrequenz (20 MHz anstatt 6,7 MHz), von den 32 Bit stehen 16 ungenutzt herum.
Die Situation ändert sich schlagartig, sobald Fließkommazahlen mit ins Spiel kommen. Dann nämlich wird die eingebaute 64-Bit-Arithmetik des Transputers genutzt.
Für die Benchmarks wurde auf der PC-Seite der schnellste C-Compiler eingesetzt, der mir zur Verfügung stand, nämlich MS-C in der Version 5. Natürlich habe ich den auch noch auf Tempo optimieren lassen, aber auch das half nichts. Das kleine Programm laut *Bild 1* lief auf dem Transputer-Board rund 200 mal schneller als auf dem 286er. Der Restfehler war übrigens bei beiden Compilern bis auf die zehnte Kommastelle identisch.
Ich habe daraufhin diverse mathematische Funktionen durchprobiert. Prinzipiell kam dabei aber immer das gleiche heraus, weshalb *Bild 2* als ein typischer Vertreter dieser Gattung reicht. Der Monoputer ist im Mittel um den Faktor 140 schneller.
Sozusagen als Praxistest habe ich nun diverse vorhandene Programme unter 3L-C compiliert, die mehr oder weniger Integer- und Float-Arithmetik gemischt einsetzen und viel mit Zeichenketten arbeiten. Hier ergaben sich Beschleunigungen um den Faktor 50 bis 80. Nun wollte ich es wissen, und ging mit den Benchmarks von Bild 1 und 2 auf eine VAX 11/780. Das war an einem Freitagabend, wo mir „Monitor" bestätigte, daß ich als einziger Anwender eingeloggt war. Der Compiler lief denn auch ungewohnt schnell, doch das war's dann auch. Der Monoputer für weniger als 3000 DM schlägt die millionenteure VAX um Längen, etwa um den Faktor 10 bis 12.

Zwei MByte RAM

Nach soviel schneller Software werfen wir jetzt noch einen Blick auf die Hardware, die das alles möglich macht.
Der Monoputer ist eine Steckkarte in langer Bauform, die aber nur einen 8-Bit-Slot benötigt. Die Karte belegt die Port-Adressen 150H bis 163H. Wenn andere Erweiterungen diese Adressen schon belegen sollten, müssen sie geändert werden. Der Monoputer hat keine DIP-Schalter oder ähnliches für diesen Zweck.
Falls Sie den Monoputer in einem alten PC mit nachgerüsteter Festplatte (nicht XT) einsetzen sollten, gibt es einen Adressenkonflikt. Für diesen Fall bietet der Hersteller einen Chip-Wechsel an, der die Port-Adresse in 300H ändert.
Die Steckkarte selbst ist mit einem T800 bestückt; daß ist der leistungsstärkste Transputer, den INMOS derzeit zu bieten hat. Dazu kommen noch zwei MByte RAM, dessen 72 IC für die Größe der Karte verantwortlich sind.

Einfach, schnell und gut

Transputer und Software arbeiteten während des Tests absolut fehlerfrei. Das System ist sehr einfach zu installieren und zu programmieren (C-Kenntnisse vorausgesetzt). Ich kenne kein anderes System, bei dem man auf so einfache und preiswerte Weise an derart hohe Rechenleistungen kommt. Gesamtnote: summa cum laude.

Literatur

[1] Die mc-Transputer-Karte, mc 4/88 bis 9/88.

Bernhard Bauer

CAD-Layout, ein spezieller Markt

Das Layout-Paket Draftsman

Die Auswahl eines CAD-Layout-Paketes ist nicht einfach: Da es weder gute noch schlechte Systeme gibt, sondern immer nur ein gutes oder schlechtes System in Bezug auf die spezielle Anwendung, sind globale Aussagen nicht seriös. Hier wird das Problem anhand eines Beispiels diskutiert. Es zeigt sich, daß die Erfahrung des Anwenders eine Rolle spielt.

Entscheidungen für ein bestimmtes System fallen durch Begrenzungen im Entscheidungsspielraum. Der Entscheidungsspielraum wird begrenzt durch:

- finanzielle Mittel
- Einsatzgebiet/Anwendung
- technische Gegebenheiten (Hardware)
- logische Gegebenheiten (Software)
- Art der Ausführung der logischen Gegebenheiten

Grundsätzlich kann man sagen, daß es Software-Entwickler im Elektronik-CAD-Bereich in der Vergangenheit schwer hatten, die Wünsche und Anforderun-

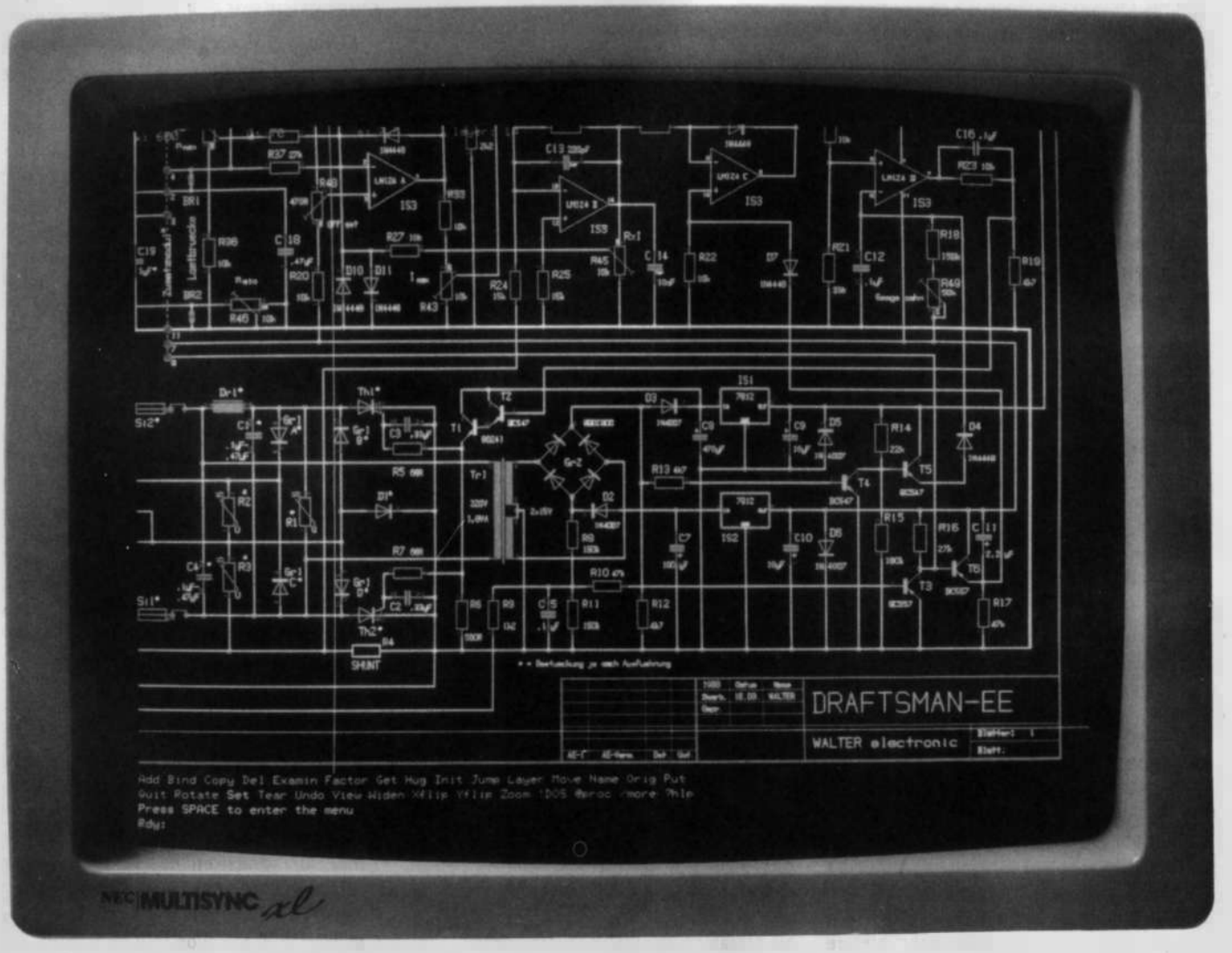


Bild 1. Ein komplexer Schaltplan im Ausschnitt

gen bzw. Verbesserungsvorschläge der Kunden (des Marktes) zu verwirklichen. Die Gründe lagen oft im Management-Bereich (Termine, Prioritäten, Verkaufen), so daß die Entwickler nicht so durften wie sie gekonnt hätten. Wahrheit war aber auch, daß Entwickler eben keine Layouter waren und so kein Gefühl von den wirklichen Bedürfnissen der Anwendung hatten. Dies gilt sowohl inhaltlich als auch für die Mensch-Maschine-Schnittstelle. Herausgekommen ist Frust bei den Anwendern und der Ärger mit den Systemen. Die Situation wurde durch die Fehler in den Systemen verschärft, mit denen sich die Anwender herumzuschlagen hatten. Der Markterfolg oder die gegenwärtige Marktbereinigung im Bereich CAD-Layoutsysteme spiegeln die Sünden der Vergangenheit wieder. Sie sind ein Indiz für eine schlauer gewordene Kundenbasis. Der Hardware-Preisverfall hat die Anwendung CAD-Layout einem breiteren Anwenderkreis zugänglich gemacht. Seit es die 386er Systeme gibt, mit den entsprechenden ADD-ONs (Grafikkarten, die Auflösungen bis 1280 × 1024 Punkten bieten, Transputerboards für schnelleres Autorouting, usw.) hat sich die CAD-Layout-Anwendung auf unterster Ebene stark verbreitert. Durch leistungsfähige Software können auf PC auch Multilayer und größere Boards erstellt werden. Heute ist die Übersicht über PC-basierte Systeme (Personal Computer – soll man das noch sagen?) auch für den Insider schwer transparent zu gestalten. Eine Empfehlung ist nicht leicht auszusprechen. Berücksichtigt doch eine Aussage über die Eigenschaften eines Systems nicht oder nur in ungenügendem Maße die Adaptierung des Benutzers an das System und somit die Gesamteinsatzfähigkeit und Effizienz der Anwendung. Es soll hier um Layoutentflechtung gehen, also ist der wichtige Punkt: Die Fähigkeiten der Layouter müssen bei der Anschaffung eines Systems mit berücksichtigt werden.
Knopfdrucksysteme, bei welchen alles von alleine geht, gibt es leider nicht. Die Vorstellung von „Schaltung rein, Layout raus" wird noch oft geäußert und noch von manchen Vertriebsingenieuren unterstützt, die da sagen: „Ach, die Maschine bedient sich ja von selbst." Grundsätzlich ist dazu zu bemerken, daß die Lösungen, die ein System dem Benutzer für seine Arbeit bietet, nie vollständig sein kann (Software wird nie fertig). Aber es gilt auch: In Deutschland werden eben andere Vorgaben an ein Layout gestellt, als es sich die Amerikaner träumen. Durch die bitteren Erfahrungen vieler Kunden mit großen Systemen hat sich eine gewisse Vorsicht etabliert, die sich in langen Entscheidungszeitspannen zwischen Beratung und Kauf eines Systems niederschlagen.
Man kann solche Anwendung wie CAD-Layout nicht über den Ladentisch verkaufen. Und ein Kunde muß darüberhinaus später bei der Anwendung hilfestellend unterstützt werden (--> kompetenter Ansprechpartner). Also ist vom Kauf eines CAD-Layout-Systems auf Grund des Preises allein dringenst abzuraten.

Draftsman, ein Low-Cost-System

Im Folgenden wird das System Draftsman beschrieben, das als Low-Cost-System gelten kann, dem Anwender in großem Umfang Freiheiten bei der Gestaltung seiner Arbeitsumgebung läßt, und verschiedene Leistungsmerkmale aufweist, die für den geübten Anwender sehr hilfreich sind. Die Freiheiten, die das System dem Benutzer bietet, bringen natürlich auch Nachteile. Die verantwortliche Handhabung des Systems liegt sehr stark beim Benutzer. Ist dieser nicht eingearbeitet und besitzt wenig Erfahrung vom Handwerk des Layoutens, kann er entsprechende Fehleingaben oder -handlungen durchführen. Gerade die flexible Vorgabe von Parametern zur Steuerung der verschiedenen Programmodule kann, wenn nicht richtig angewendet, zum Nachteil wirken: Wo viel gedreht werden kann, kann auch viel verdreht werden.
Dies ist prinzipiell eine Grundsatzentscheidung bei der Anschaffung eines CAD-Layoutsystems: Wie groß sollen die Freiheiten eines Anwenders sein? Hier treffen Philosophien aufeinander. Die Entscheidung hängt stark von den Umgebungsverhältnissen ab, in denen das System eingesetzt wird. Großfirmen wollen oft eine mehr reglementierte Arbeitsweise, da das System auch von mehreren Mitarbeitern bedient werden soll. Die spezielle Anpassung an eigene Verhältnisse ist dann auch eher bei kleineren Firmen oder bei Experten zu finden. Draftsman ist deshalb nicht für jemanden gedacht, der ein schlüsselfertiges, einfach zu bedienendes System erwartet, sondern für jemanden, der sich mit dessen flexiblen Möglichkeiten auseinandersetzen will und sich seine eigene Arbeitsumgebung und seinen eigenen Zugang zum System definieren möchte.
Draftsman fängt semantische Fehleingaben nicht von vorneherein ab, sondern überläßt dem Anwender die Verantwortung, wie er mit dem System umgeht. Er muß wissen, was er tut (sonst: Warten auf KI-Systeme!). Umfangreiche Checks gehen auf Kosten des Software-Overheads und verlangsamen das Gesamtsystem. Durch leistungsfähigere Hardware kann dieser Nachteil verringert werden. Umfangreiche Software, viel Overhead, teuere leistungsfähige Hardware, Verantwortung eher beim System, das kennzeichnet die eine Philosophie. Optimierte Software mit möglichst geringem Overhead, die auf einfacher Hardware läuft und dem Anwender die Verantwortung überläßt, das ist die andere Philosophie bei CAD-Layoutsystemen. Und dieser folgt Draftsman.

Das Hardware-Umfeld

Draftsman läuft auf den IBM-Kompatiblen, vom PC bis zu den schnellen 386er Systemen. Ein Coprozessor beschleunigt die Arbeitsgeschwindigkeit und ist zu empfehlen. Ein Rechner mit 8 MHz Takt ist zudem schon recht langsam. Empfohlen werden mindestens 10 MHz. Die Grafik-Standardkarte ist EGA, es können auch ATI Wonder und Genoa eingesetzt werden. Mit CGA und Monochrom sollte nicht gearbeitet werden. Eine zusätzliche Grafikkarte mit einem hochauflösenden Monitor für einen schnelleren Bildaufbau ist ebenfalls für größere Arbeiten dringend zu empfehlen. Eine Speichererweiterung nach Extended Memory System (EMS) ist schon für mittelgroße Schaltbilder unbedingt zu empfehlen. Sonst ist der Rechner nur mit Swapping beschäftigt. Ein Cache-Speicher wäre eine Alternativ-Lösung, der Preis wird entscheiden.

Der Editor

Das System besteht aus einem universellen Grafikeditor, der für alle grafischen Eingaben benutzt wird. Die unterschiedlichen Einstellungen für die Anwendungsbereiche Symbol-, Schaltbild- und Layouterstellung können interaktiv eingegeben oder zusätzlich als Parameterdatei gehalten werden.

- Netzlisten und Routing-Prüfroutinen (Off-Line) geben dem System durchgängige Sicherheit
- Netzlisten, Materialliste, Partsliste sind die Standardausgaben
- es gibt ein leistungsfähiges Plazierungsprogramm mit halbautomatischer Vorgabe der zu plazierenden Bauteile und mit der Option Parameterwerte zu ändern.
- es gibt ein leistungsfähiges Routing-Programm mit der Fähigkeit eines Punkt zu Punkt Autorouters.
- In zwei Ausführungen ist ein Autorouter für bis zu 32 Lagen automatisch zu routende Leiterplatten mit Parametersteuerung und Clean-Up Lauf zu haben.
- Ausgaben für Pen-Plotter, Gerber Photoplotter und Excellon Bohrmaschine bilden die Schnittstelle zu CAM.

- Eine umfangreiche Bibliothek wird mitgeliefert. Sie unterliegt den üblichen amerikanischen Normen.
- Das System verwendet Checks bei der Netzlisten- und bei der PCB-Erstellung.
- Informationsfelder.

Eine fortschrittliche Eigenschaft des Systems ist die Zuordnung von Parametern für die Layoutgestaltung schon im Schaltbild. Die Parameter WIDTH und VIA werden benutzt, um einem bestimmten Netz eine bestimmte Tracebreite im Layout vorzugeben. Zusätzlich kann auch der Typ einer Durchkontaktierung vorgegeben werden. Das Check-Programm und der Autorouter berücksichtigen diese Parameter. Der Benutzer hat die Möglichkeit sog. globale Pins zu erstellen. In dem Symbol sind diese einem globalen Netz zugeordnet und müssen nicht in der Grafik als Pin erscheinen (VCC,GND).

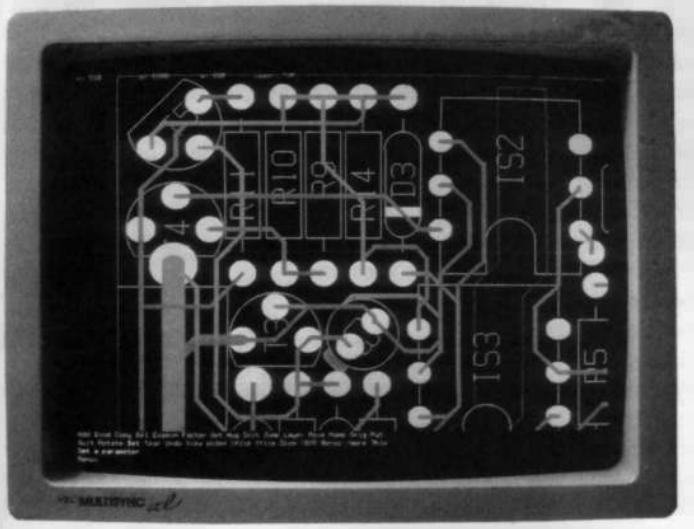


Bild 2. Ein Ausschnitt aus einem Layout

Das Informationssystem

Verschiedene Full-Screen-Menüs werden als Informationsausgabe oder auch als Eingabe-Werkzeug verwendet. Alle gesetzten Parameter und die System-Konfiguration können am Bildschirm dargestellt werden. Unter „Gesetzte Kommandos" kann der Benutzer sein Working-Environment ansehen. Unter „Workspace Statistics" erscheint der augenblickliche Dateiname und eine Angabe über die Anzahl der benutzten Symbole und Elemente. Unter „System Konfiguration" werden die Belegung des Speichers, Plotter-Einstellungen usw. ausgegeben und die Zuordnung der Lagen zu den Datenelementen und Farben. Außerdem die Funktionstastenbelegung mit Makrokommandos und der Suchpfad für die einzelnen Bibliothekelemente in den verschiedenen auch vom Anwender frei zu belegenden Dateiverzeichnissen. Es gibt einen HELP-Screen, in dem alle Abkürzungen der Befehle und weitere Informationen enthalten sind. Es gibt eine Aperturetable für die Zuordnung zu einem Photoplotter. Der Anwender bekommt so eine gute Übersicht über die gesetzten Parameter und Einstellungen. Dem Anfänger bietet das System eine Menüauswahl auf dem unteren Ende des Bildschirmes. Diese Hilfe ist abschaltbar und wird nach einiger Zeit nicht mehr benötigt. Die beiden Menüzeilen beinhalten nacheinander alle Hauptkommandos, nach Selektieren mit der Maus oder den Cursortasten erscheint die zweite Kommandoebene mit den Attributen. Selektiert man ein Attribut, erscheint die nächste Ebene der Attribute. Das System gibt als Hilfe immer den nächsten Schritt an, sei es die Wahl eines Kommandos, Attributes oder die Positionierung des Cursors, hierbei sogar den Anfangspunkt, die Zwischenpunkte, sowie auch den Abschluß. Ein Side-Screen-Menü, wie man es von den meisten anderen Systemen kennt, ist hier nicht verwirklicht. Die Eingabe der Kommandos erfolgt durch Eingabe der Kommandoabkürzungen über die Tastatur.

Das Layerkonzept

Draftsman stellt dem Benutzer 64 Lagen zur Verfügung. Je nach Applikation sind bestimmte Lagen für bestimmte Datenelemente (wie z. B. Symbole, Leitungen, Pins, ...) vordefiniert. Der Benutzer kann aber frei über alle Lagen verfügen und sie nach seinen Bedürfnissen definieren. Einer Lage wird jeweils eine Farbe zugewiesen. Dies erfolgt aus einer Farbtabelle mit 16 Einträgen. Ebenso wird ein Datenelement einer Lage zugeordnet. Diese Zuordnungen sind über die Tastatur mit dem entsprechenden Kommando oder mit Hilfe eines Full-Screen-Menüs durchführbar. So ist auch die Anzeige der aktuellen Zuordnungen über dieses Menü abrufbar [SET COLOR SCREEN SCOL]. Die Handhabung der Layer-Zuordnung ist einfach. Einer Lage wird eine Farbe zugeordnet [SCOL Lage =Colorcodenr.]. Elemente werden den entsprechenden Lagen zugeordnet [SL Element Lagennr.]. Somit wird eine farbliche Unterscheidung der Elemente und die getrennte Handhabung dieser Elemente (Ein-Ausschalten der Lagen) erreicht. Weiterhin können einzelne Elemente oder Segmente von Linienelementen auf eine andere Lage umgelegt werden [LL neue Lagennr.]. Die Zuordnungen Lage/Farbe ist in einem Full-Screen-Menü wie auch in der Kommandozeile gegeben.

Die Dateien: Alle in ASCII

Alle vom System erstellten Dateien sind vom Typ ASCII und vom Anwender auch direkt modifizierbar. Für die verschiedenen Programme des Systems kann man Parameter voreinstellen. Diese werden in getrennten Dateien gehalten und steuern die zugehörigen Programme entsprechend. So ist für jeden Systembereich ein einfaches Erstellen und Ändern der Programmverhaltensweisen gegeben. Das System bietet die Fähigkeit, Kommandos oder Kommandosequenzen als Makro zu definieren und auf eine Funktionstaste zu legen. Diese Makrofähigkeit wird eingeschränkt auf 70 Zeichen pro Makro und auf die drei Belegungsmöglichkeiten der Funktionstasten (F1-F4 direkt, CTRL-F1-F10 und ALT-F1-F10) sowie die drei Maustasten. Insgesamt 27 Tasten. Die Fähigkeit, diese Belegung in einer Datei abzuspeichern und nach Belieben aus dem Editor aufzurufen, ist leistungsfähig und gibt dem Benutzer die Möglichkeit, häufig benutzte Kommandos oder spezielle Sequenzen nach Bedarf zu nutzen. Ein Nachteil ist die Darstellung der Kommandos, die bei den Funktionstasten als Papierstreifen über den Tasten erfolgen muß. Vorteil ist die farbliche Gestaltung nach den Bedürfnissen des Benutzers und die Möglichkeit, auch icon-artige Symbole an Stelle von Text darzustellen. Diese Papierstreifen muß man entsprechend der Belegung von Fall zu Fall wechseln. Von Vorteil ist hierbei, daß der gesamte Bildschirm für die Grafik nutzbar ist. Eine Funktionstaste kann eine andere aufrufen, es kann eine hierarchische Kommandostruktur implementiert werden. Die Funktionstastendateien können mit dem Editor des Systems oder auf DOS-Ebene mit einem Texteditor oder Textverarbeitungssystem (z. B. Wordstar, Siedekick) bearbeitet werden. Dokumentation und Änderungen sind so leicht zu bewerkstelligen. Die Funktionstastenbelegung kann auch auf ein Tablett übertragen werden. Durch die Flächenbelegung können die zu einer Datei gehörenden 24 Funktionstasten (F1-F4, CTR-F1-10, ALT-F1-10) direkt angeordnet werden. Parallel

dazu können somit auch mehrere Dateien dargestellt werden. Der Benutzer kann sich so seine grafische Bedieneroberfläche nach seinen Gegebenheiten erstellen.

Die Startup-Datei und die Handhabung

Das System benutzt eine dreistufige, hierarchische Startup-Prozedur. Für die Anpassung an die verschiedenen Vorgaben des Anwenders von seiner Aufgabenstellung her und für die getrennte Handhabung bei der Symbol- Schaltbild- und Layout-Erstellung kommt diese Systemstruktur dem Anwender sehr entgegen. Zunächst wird ein Programm gestartet, das die Bibliothek-Suchpfade und die Hardware-Peripherie setzt, die Maustastenbelegung und den Aufruf einer anderen Prozedurdatei durchführt. Dort sind die Einstellungen für das Arbeiten unter Symbol- Schaltbild- und Layoutumgebung enthalten. Eine Grundbelegung der Funktionstasten wird ebenfalls in dieser Systemphase geladen. Das Arbeiten mit dem System erfordert eine Denkweise, die zu Anfang sicher nicht jedem leicht fällt. Normalerweise funktionieren CAD-Systeme nach der Methode „Was, wie, wo":

1. Hauptkommando selektieren
2. evtl. Unterkommandos selektieren (definieren von Attributen)
3. Selektionspunkt für zu modifizierende Elemente oder Startpunkt für die Erstellung neuer Elemente wie z. B. „Linie setzen".

Bei Draftsman ist ein anderer Zugang zum System in der Handhabung schon aus der Beschreibung der Kommandosyntax erkennbar. Das liegt am historischen Hintergrund des Systems. Es war auf die Bedienung über die Tastatur ausgerichtet und bietet jetzt auch Handhabung über die Maus. Die ursprünglichen Kommandostrukturen sind geblieben. Wenn zum Beispiel ein Kommando gewählt wird, bezieht es sich auf eine vorher eingestellte Cursorposition. Das bedeutet, daß man vor der Eingabe der Kommandos die Position für die Aktivität am Bildschirm kennen muß. Positionen werden mit der Maus festgelegt, wobei ein Raster unterlegt werden kann. Kommandos werden über die Tastatur gewählt. Danach wird mit den Cursor-Tasten oder mit der Maustaste das Datenelement erstellt, z. B. eine Linie. Der Abschluß eines Kommandos wird wieder mit einer Maustaste oder der ENTER-Taste gegeben. Das Vorplazieren des Cursors und anschließende Eingeben des Kommandos erfordert eine Disziplinierung der Arbeit. Die Handhabung mit der Maus ist gewöhnungsbedürftig. Die Darstellung eines Gummibandes vom letzten Punkt auf dem Bildschirm zur Cursorposition entfällt. Das Führen eines Elementes am Cursor ist in Echtzeitdarstellung sowie auch als umschließendes Rechteck nicht möglich.

Feststellungen

Die Eingabe der Kommandos über die Tastatur ist nicht so schwer, wie anfangs vermutet. Die Auswahl der Kommandoabkürzungen und deren Attribute sind auf die Anfangsbuchstaben beschränkt und auch einleuchtend abgekürzt. So ist zum Beispiel das Definieren einer Lage für elektrische Verbindungen mit SLW Nr. (Set Layer Wire) abgekürzt, MS bedeutet Move Symbol und RS90 bedeutet Rotate Symbol 90 Grad. Der Kommandosatz ist hierarchisch aufgebaut: erster Buchstabe = Hauptkommando, zweiter = Attribut, dritter = Attribut oder Wert zum Attribut). Den Kommandosatz hat man sehr schnell auswendig gelernt und kann ohne das Menü direkt arbeiten. Da das System bei einer Kommandoeingabe ohne Wertangabe den geltenden Wert als Information ausgibt, ist auch eine schnelle Abfrage der augenblicklich geltenden Parameter möglich. Draftsman bietet dem Benutzer eine Vielzahl von Jump-Befehlen mit denen zu einem bestimmten Datenelement gesprungen werden kann. Datenelemente sind Symbol, Linie, Wire, Bus, Text, Nullpunkt der Bildschirmdarstellung, usw. Das Datenelement kann einen Namen besitzen. Dieser Name kann zusätzlich mit eingegeben werden (z.B: ein bestimmtes Netz). Bei Linienelementen kann zum Mittelpunkt oder zum anderen Ende gesprungen werden. Natürlich kann auch ein Koordinatenpunkt oder ein Name eines Symbols vorgegeben werden. Liegt das Zielobjekt außerhalb der augenblicklichen Bildschirmdarstellung, wird nach dem vom System selbständig ausgeführten Panning automatisch ein neues Bild aufgebaut. Mit bestimmten Tasten (CTRL-X) kann, während ein Kommano aktiv ist, zu einem bestimmten Element gesprungen werden. Das System besitzt die Fähigkeit des Autopannings.
Prinzipiell ist zu sagen, daß reine Mausbenutzer-Fans es sicher schwer haben werden, Draftsman zu bedienen. Andererseits ist es nach einer entsprechenden Einarbeitung in ein CAD-System immer schneller, oft auch einfacher, beidhändig zu arbeiten, da die Cursorbewegungen zum Side-Screen-Menü oder das Durchhangeln über mehrere Pop-Up-Menüs (bis man die gesamte Kommandosequenz selektiert hat) wegfallen. Beidhändig zu arbeiten ist kein Nachteil, sondern höchstens ungewohnt.

Die Flexibilität

Parametereingaben können in dem entsprechenden Programmodul interaktiv oder auch in Dateien vorgegeben werden. Verschiedene Zuordnungen können während der Symbolerstellung, Schaltbilderstellung und später erfolgen. Die Zuordnung der Gehäuseform kann im Symbol, im Schaltbild und bei der Plazierung im PCB vorgenommen werden. Die Zuordnung der Globalen Pins kann im Symbol, oder im Schaltbild erfolgen. Repetierendes Handling ist mit der SPACE Taste gegeben. Damit ist gemeint, daß zum Beispiel mehrere XY-Positionen eines neu zu plazierenden Elementes sozusagen probeweise gesetzt werden – und zwar nacheinander durch Drükken der Maustaste eins, bis durch Drücken der zweiten Maustaste (Abschluß) die entgültige Position bestätigt wird. Ein Anwender kann auch jederzeit aus dem Editor auf die DOS-Ebene zugreifen und dort arbeiten. Aus dem Editor können jederzeit beliebige Kommandodateien aufgerufen werden. Daß die Handhabung des Systems sehr schnell ist, zeigt sich dann, wenn der Benutzer die Befehle auswendig kann und das Menü ausschaltet.

Die Bibliothek

Großen Einfluß auf die Arbeit mit CAD-Anlagen hat der Umgang mit der Bibliothek. Hier ist der Umfang der mitgelieferten Symbol- und Bauteilbeschreibungen, die Art der Darstellungen (amerikanisch, deutsch oder DIN) als auch die Handhabung (Zugriff auf diese Daten) ein Kennzeichen für den Aufwand, den der Benutzer nachher in der Praxis hat. Der konzeptionelle Entwurf einer Bibliothek und die daraus folgende Verteilung der dem Benutzer überlassenen Arbeit mit der Bibliothek (beim Selektieren von Symbolnamen) ist für die praktische Arbeit von Bedeutung. Zum Beispiel kann der Name für einen Widerstand nur ein R sein und die dazugehörige Grafik erscheint. Die Eingabe des „Reference Designators", des Wertes und anderer Attribute muß für jedes Symbol im Schaltbild durch den Benutzer erfolgen. Die Unterscheidung zwischen den Werten eines Bauteils kann aber auch in der Bibliothek im Namen erfolgen. Diese hat dann entsprechend viele Symbole und Namen, die je nach den Unterscheidungsmerkmalen sehr lang werden können, wenn Attribute wie Toleranzen, Belastbarkeit, usw. mit berücksichtigt werden sollen. Der Benutzer hat dann einen langen, oft schwer zu merkenden Namen einzugeben. Das System sollte hier eine entsprechende Be-

```
DRAFSTMAN-EE PARTS LIST EXAMPLE                          Page: 1
Schematic: CASE1.SCH                                     Date: Tue May 10 1988
Sorted by: INCREASING FILENAME                           Time: 02:11:27

          PART              PART          REF.      _____LOCATION ON SCHEMATIC_____
ITEM IDENTIFICATION      FILENAME         NAME      DATAFILE      PG      X       Y
==== ================ ============== ======== ============== == ====== ======
   1 (none)           bsize          (none)   CASE1.SCH       1      0      0
   2 01-123-4455      ls00a          U6                       1    540    240
   3                  ls00b          U6                       1    570    360
   4                  ls00c          U6                       1    410    580
   5                  ls00d          U6                       1    340    130
   6 01-044-0300      ls02a          U4                       1    410    480
   7                  ls02c          U4                       1    340    210
   8 44-888-0012      ls21a          U8                       1    410    270
   9 03-443-19AA      ls245          U1                       1    770    470
  10                                 U3                       1   1010    470
  11                                 U5                       1    770    190
  12                                 U7                       1   1010    190
  13 01-432-7755      ls74a          U2                       1    480    110
  14                  ls74b          U2                       1    560    500


DRAFTSMAN-EE BILL OF MATERIALS EXAMPLE                   Page: 1
Schematic: CASE1.SCH                                     Date: Tue May 10 1988
Sorted by: INCREASING PART ID                            Time: 02:11:49

          PART                                                    UNIT      TOTAL
ITEM IDENTIFICATION              DESCRIPTION                QUAN.  COST      COST
==== ================ ================================== ===== ======= ========
   1 01-044-0300      LS02 Quad 2-input NOR gate             1    0.65      0.65
   2 01-123-4455      LS00 Quad 2-input NAND gate            1    0.52      0.52
   3 01-432-7755      LS74 Dual D-type flip-flop             1    0.74      0.74
   4 03-443-19AA      LS245 Octal transceiver, 3-state       4    0.97      3.88
   5 44-888-0012      LS21 Dual 4-input AND gate             1    0.42      0.42
                                                          ===== ======= ========
Total:                                                      8    0.78      6.21

Macros with no Bill of Materials data: 1
```

Bild 3. Eine Stückliste und die zugehörige Materialliste

nutzeroberfläche bieten, in der man über Menüzeilen das Symbol erhalten kann. Bei digitalen oder analogen ICs, bei denen sich mehrere identische Untereinheiten (Gatter, Operationsverstärker) in einem Gehäuse befinden, haben wir eine ähnliche Situation. Die Untereinheiten müssen unterschieden werden. Das erfolgt mit einem Anhängsel A, B, C oder D. Diese Zusatzinformation kann entweder durch den Benutzer im Schaltbild eingegeben werden oder in den Symbolnamen integriert werden, wobei dann der Umfang der Bibliothek vergrößert wird. Die Zuordnung zum physikalischen Gehäuse wird bei Draftsman ebenfalls als Parameter eingegeben. Es kann zwischen bedrahteten Bauteilen und SMDs unterschieden werden. Bei der Erstellung einer Materialliste wird auf die Daten in den Informationsfeldern zugegriffen. Sollen alle Technologien (TTL: Std, S, LS, AS, ALS, HC, HCT, ACL) berücksichtigt werden, müssen ebensoviele Dateien (Grafik + Infofeld) erstellt werden. Sollen diese Technologien für verschiedene Gehäuseformen (DIP, SO, Chip Carrier, Flat Pack) vorhanden sein, multipliziert sich das. Weiterhin kann noch zwischen Keramik- und Plastikausführung unterschieden werden. Bei unsachgemäßer Anwendung wird die Bibliothek sehr schnell wachsen. Die Erstellung nur derjenigen Symbole, die auch wirklich gebraucht werden, ist sinnvoll.

Die Bibliothek enthält für Gatterbausteine (74LS00) mehrere vorgegebene Symbole. Zunächst ein Grundsymbol ohne Pinnummern (Grafik) sowie die anderen Gattersymbole mit Pinnummern mit den entsprechenden Anhängseln A,B,C,D für die vier Untereinheiten des Bauteils. Ein Anwender muß also die Pinnummern und den erweiterten Partnamen nicht im Schaltbild eintragen, was die Bibliothek aufbläht. Vorteil ist aber, daß ganze Gatter mit den richtigen Zuordnungen neu plaziert werden können und die Fehlerquelle bei der Eingabe von Pinnummern entfällt. Diese Vorgehensweise ist zu empfehlen, zumahl auch Änderungen schnell und sicher durchgeführt werden können. Datenelemente sind alle Primitiven, zum Beispiel Linie, Kreis, Draht, Route, Text, usw. Das System unterscheidet elektrische und nichtelektrische Elemente. Ein Librarymanager wird vom deutschen Distributor als Eigenentwicklung zum System mitgeliefert. Er definiert den Zugriff auf die Bibliothek in der Weise, daß der Anwender die Symbole und Bauteile als Dateinamen aus den verteilten Directories dargestellt bekommt und durch Anklicken mit der Maus selektieren kann. Zusatzinformationen zum Symbol oder Bauteil können abgerufen werden. Die selektierten Dateien können dann in eine lokale Bibliothek kopiert werden. In dieser Bibliothek können Symbole oder Bauteildefinitionen nach Anwendererfordernissen geändert werden, ohne die Standardbibliothek zu tangieren. Außerdem kann diese lokale Bibliothek dann mit Schaltbild, Layout und allen anderen Dateien als komplettes Projekt ausgelagert werden und so bei späteren Erweiterungen oder Modifikationen unabhängig von Symbol- und Bauteiländerungen in der Standardbibliothek sicher weiterverwendet werden.

Symbole und Bauteile

Das System umfaßt 1791 Symbole und beinhaltet die Technologien : ALS, AS, LS, ECL, HC, HCT, verschiedene Speichersymbole, diskrete Symbole, Pads und Packages.

Draftsman benutzt das Ebenenkonzept für die hierarchische Symbolerstellung. Es ist eine Schachtelung bis zu 20 Ebenen tief erlaubt. Die Symbole beinhalten die graphische Darstellung, die elektrischen Verbindungspunkte für die Netzliste und Parameter, welche für das Schaltbild die entsprechenden Werte enthalten: Ref.Des., Wert, Part-Nr., Angaben wie Toleranz, Belastbarkeit, ... Parameterangaben für die Layoutgestaltung sind bei diesem System ebenfalls auf Symbol- und Schaltbildebene möglich. Vorteil dieses Konzeptes ist die schnelle Modifizierbarkeit. Nachteil ist der größere Umfang der Bibliothek. Es gibt keine Parts-Datei, aus der auf die Symboldatei und PCB-Datei referenziert wird. Diese Zuordnungen sind symbolgebunden. Bei der Symbolerstellung werden diese Referenzen als Parameter mit weiteren Daten in den Informationsfeldern des jeweiligen Symbols eingetragen (zum Beispiel: Name des Symbols: U1, Description: 74LS00, PartID: firmenspez. Nummer, Gehäuse des physik. Bauteils, ein Cost-Parameter, globale Pins (VCC,GND)). Bei der Neuerstellung können vorhandene Symbole kopiert und modifiziert werden. Von Vorteil bei Draftsman ist das Kommando FACTOR, welches eine

Vergrößerung oder Verkleinerung eines Datenelementes um einen einzugebenden Prozentwert bewirkt.

Schaltbilder

Ein Schaltbild kann bis zu 100 Seiten Umfang besitzen. Verbindungen über Seitengrenzen hinweg werden wie üblich über die Netznamen verwaltet. Abblockkondensatoren müssen nicht eingegeben werden, sie werden über das Netzlistenprogramm (Eingabe der Netznamen für VCC, GND) auf Grund des Common-Pin-Konzeptes extern verwaltet. Beim Rotieren von Symbolen bleibt der zugehörige Text immer von zwei Seiten lesbar. Die besondere Behandlung der gedrehten Texte wird somit vermieden. Änderungen der Werte in den Informationsfeldern der Symbole sind auch im Schaltbild gegeben. Aus Eindeutigkeitsgründen sollten solche Änderungen aber auf Experimente beschränkt bleiben.

Der Output

Die Informationen für Materiallisten und Stücklisten werden aus den Informationsfeldern der Symbole entnommen. In der Materialliste werden zum Beispiel die Cost-Parameter der einzelnen Bauteile addiert und auch als Gesamtsumme ausgegeben. Das Netzlistenprogramm ist zweistufig. Es können Netzlisten von bis zu 100 Schaltbildseiten erstellt werden. Spezielle Daten aus den Informationsfeldern der Symbole werden für die spätere Layouterstellung mit berücksichtigt.

Die Netzlisten: vektororientiert

Netzlistendateien sind vektororientiert, in einem freien Format mit variabler Länge im ASCII-Format organisiert. Die Verbindungen innerhalb eines Makroelementes (im Schaltbild eine Bauteilgruppe) werden mit berücksichtigt. Bei der Erstellung eines Schaltbildes ist es von Vorteil, Gruppen bewegen, rotieren und so als Einheit behandeln zu können. Dabei müssen die makrointernen Verbindungen natürlich mit berücksichtigt werden. Checks, die das Programm ausführt, helfen bei der Fehlerbeseitigung. Die Fehlermeldungen erscheinen in der Netzlistendatei. Bei größeren Netzlisten ist das unhandlich, muß doch dann die gesamte Datei ausgedruckt werden. Batchdateien unterstützen die Zusammenfassung der endgültigen Netzliste aus den Listen zu den einzelnen Seiten. So kann auch bei Änderungen der Arbeitsaufwand minimal gehalten werden.

PCB-Layout, Parametereinstellung und Checks

Überprüft wird der Abstand von Trace zu Trace, Pads, Vias, Exclusion Areas, Board Outline sowie von Pad zu Pad, Via zu Via und Pad zu Via. Dabei können die Pads und Vias neben Kreisen und Rechtecken beliebige Formen aufweisen. Auf unvollständige Verbindungen und Kurzschlüsse durch Leiterbahnen wird abgeprüft, wobei der Ort eines Kurzschluss nach XY-Koordinaten gekennzeichnet wird. In der neuen Version wird der Fehlerort auch grafisch am Bildschirm markiert. Stichleitungen (Hangers) bringen eine Warnung. Dies ist sinnvoll, da z. B. eine Stichleitung, die als Masseabschirmung parallel zu einer Leiterbahn geführt werden soll, nur an einem Ende an Masse angebunden werden darf. Checks und Checkparameter können interaktiv oder durch Parameter in der Kommandozeile beim Aufrufen des Checkprogramms vorgegeben werden, was gezielte Überprüfung möglich macht. Für verschiedene Layouts können die entsprechenden Checkvorgaben erstellt werden. Der Autorouter als auch der halbautomatische Punkt-zu-Punkt-Router prüfen direkt auf das „Rat's Nest", während der Check sich an der Netzliste orientiert.

Placement

Das Placementprogramm beinhaltet einen Automatismus, der es dem Anwender gestattet, die Reihenfolge der angebotenen Bauteile nach seinen Vorgaben über Parameter zu definieren. In der oberen Zeile am Bildschirm wird dazu eine dreiteilige Information ausgegeben. Sie beinhaltet den Ref.Des.-Namen, Dateinamen und das Gehäuse eines Bauteils. Mit den ±-Tasten kann der Anwender durch die Bauteilliste scannen. Es können mit Wildcardzeichen (Fragezeichen wie von DOS her bekannt) auf Namen operiert werden. Beispiel: SMASK N=U??? bedeutet, daß alle Ref. Des.-Namen, die mit U beginnen und dreistellig sind, nacheinander in aufsteigender Reihenfolge angeboten werden. Die Plazierung muß in einer Sitzung nicht zu Ende geführt werden, das System speichert die Plazierungsliste jeweils auf dem aktuellen Stand. Bypass-Kondensatoren können hier frei plaziert werden.

Der HUG-Algorithmus

Es gibt einen halbautomatischen Punkt-zu-Punkt-Router (HUG) Dieser Programmteil routet automatisch eine Leiterbahn von einem Punkt, der vom Anwender vorgegeben wurde, bis zu der Cursorposition. Diese kann entweder der zweite Punkt einer Verbindung sein oder ein Zwischenpunkt, der dem Router vorgegeben wird, weil er in dieser Richtung routen soll. So kann der Anwender die Verlegung der Leiterbahn beeinflussen. Mit den steuerbaren Parametern hat er eine komplette Kontrolle über die Art der Verlegung. So kann er zum Beispiel den Parameter, der das Vorausschauen in der Routingrichtung beeinflußt, verändern. Der Parameter besagt, wie weit vor einem Hindernis die Routingrichtung geändert werden soll. Ein kleiner Wert bedeutet hier ein möglichst enges Umfahren des Hindernisses. Der Effekt ist in der Geradlinigkeit der Leiterbahnen zu finden. Entweder viele kleine Stufen oder langgezogene Bahnen. So etwas ist für Experimente aber auch für Experten, die sich mit den Routingproblemen auskennen eine leistungsstarke Hilfe. Allerdings routet er nur auf einer Ebene. Der Anwender muß durch Makros und Funktionstastenbelegung sich die entsprechenden Kommandos für das effektive Arbeiten selbst definieren. Der Algorithmus berücksichtigt auch die in der Parameterliste vorgegebenen Sicherheitsabstände und andere Vorgaben durch das Parametersetting. Dabei braucht die Leiterbahn nicht in echter Breite am Schirm sichtbar zu sein. In Bezug auf den schnellen Bildaufbau können auch die Bahnen als Strich dargestellt werden. Geroutet wird unter 45 Grad. 90 Grad müssen direkt vorgegeben werden. Es können leider nicht alle Parameter vom Editor aus geändert werden, der Look-Ahead-Parameter aber doch. Das versehentliche Routen auf ein anderes Netz (anderes Pad angepickt) führt der Algorithmus nicht durch. Er meldet sich mit der Message: Not a Co-Net. Da der Router grundsätzlich nicht rastergebunden arbeitet (1-mil-Raster), können z.B. keine 50-mil-Abstände verwirklicht werden. Der Router schmiegt die neuerstellten Bahnen an die bereits verlegten nach der Einstellung des Abstandsparameters an. Eine Forderung wäre hier, daß man bei Bedarf ein Raster definieren kann. Verbotene Zonen werden ebenfalls berücksichtigt, so daß Routinggrenzen entsprechend vorgegeben werden können. Der erfahrene Anwender hilft sich oft durch plazieren solcher Bereiche, um sie später wieder zu entfernen. So kann er das Routingergebnis steuern. Diese Vorgehensweise ist von den Vorgaben für Autorouter her übernommen. Die Vorgabe der Leiterbahnbreite von der Netzliste her wird bisher leider nicht im HUG-Algorithmus berücksichtigt. Manuelles Routing ist auch unter Umgehung von gesetzten Parametern gegeben. Der Anwender hat hier

Freiheiten, die ihm auch das Leben schwer machen können. Er kann Bahnen frei (unabhängig von der Netzliste) verlegen, löschen und an andere Pads verbinden. Neue Airlines (Rat's Nest) können eingefügt werden, andere gelöscht werden. Leider löscht das Programm die Airlines nach Fertigstellung der Bahnen nicht automatisch, so daß diese interaktiv sehr vorsichtig, zeitaufwendig und fehleranfällig beseitigt werden müsssen. Auch behindert das Vorhandensein dieser nun überflüssigen Airlines die Übersicht.
Hier ist eine große Fehlerquelle vorhanden, die bei größeren Boards leicht zu Problemen führen kann. Sind erst einmal Fehler vorhanden, ist es sehr zeitaufwendig, diese zu lokalisieren und zu beheben. Versuchsrouting, d.h. eine Leiterbahn frei erstellen, bringt ebenfalls automatisch eine Airline mit sich, die nach Löschen der Bahn am Schirm verbleibt. Sie muß einzeln gelöscht werden. Hier sind noch Verbesserungen notwendig. Verlegte Leiterbahnen können als Pattern kopiert werden. Das ist für Busstrukturen (Memory Bank) sehr von Vorteil. Nachteil ist das umständliche Definieren dieser Bahn als Makroelement und nach kopieren wieder das auflösen (flachmachen) aller kopierten Makros. So geht der Vorteil des schnellen, unproblematischen Verlegens wieder dahin. Bauteilgruppen können verschoben werden. Augenblicklich funktioniert das nur einmal. Auch Kopieren und Löschen von Bauteilgrupen ist einer späteren Release vorbehalten. Manuell eingegebene Bahnen werden vom Autorouter berücksichtigt und nicht wieder neu verlegt. Das ist aber auch eine zwingende Vorgabe an ein System. Die Verbindungen müssen immer von Pad/Via zu Pad/Via verlaufen. Eine Bahn stoßartig an eine andere Bahn (T-Stück) legen, das kann nur mit einem Trick erreicht werden. Die Darstellung der Leiterbahnen kann auf drei Arten gewählt werden. Als Strich, wegen des schnelleren Bildaufbaus, mit der echten Breite als Rechteckelement ungefüllt und mit Füllung.

Autorouting

Der Autorouter ist unterbrechbar. Daher kann manuelles Routen jederzeit eingeschoben werden. Alle Parameter, die für das manuelle Routen gelten, sind auch beim Autorouten gültig. Darüberhinaus müssen weitere Parameter gesetzt werden, wie z.B. die max. Abweichung von der Vorzugsrichtung auf einer Lage, Routergrundraster, Anzahl der Lagen, Vorzugsrichtung für eine bestimmte Lage, orthogonales odem diagonales Routing. Für die Datensicherheit existiert ein Parameter, der es erlaubt, eine Zeitspanne zu definieren, nach der periodisch das System den aktuellen Stand der Layoutdaten auf Platte sichert. Die Auflösung des Routers beträgt 1 mil. Die Vorgaben bezüglich der Routingebenen ist nicht durch paarweises Routen eingeschränkt, es werden alle 32 Ebenen parallel geroutet. Draftsman routet möglichst vollständig auf den äußeren Lagen ehe er auf die Innenlagen ausweicht. Geroutet werden kann unter allen Winkeln. Das ist für spezielle Leiterplatten (z.B. im Automobilbau, Tacho-Boards) notwendig. Für Hochfrequenzanwendungen oder hochgetaktete Prozessorboards kann auch mit verrundeten Ecken geroutet werden. Dies ist eine Option die selten zu finden ist. Weiterhin kann innerhalb eines Leiterbahnbereiches die Bahn auch verschmälert werden. Beim Aufrufen kann vorgegeben werden, ob im Clean-Up Lauf nur Bahnen oder auch Vias behandelt werden sollen. Wenn eine Minimierung der Vias gefordert wird, kann der Router auch beliebig gegen die Vorzugsrichtung arbeiten. Je nach Firmennormen oder Vorgaben vom Kunden ist eine mehr oder weniger restriktive Handhabung der Verletzung der Vorzugsrichtung gegeben. Das bedeutet, daß die Fähigkeiten des Routers vielleicht nicht voll zur Geltung kommen können.

Der Router zu Draftsman ist kein Rip-Up/Retry Router. Es wird aber der Bartels-Autorouter angepaßt werden, der dieses Konzept enthält. Als deutsches Produkt ist dieser auf Grund seiner Leistungsfähigkeit (lokales Rip-Up/Retry-Backtracking) zum Standardrouter geworden. Der Router von Draftsman benutzt drei Phasen. Im ersten Lauf benutzt er den gleichen Algorithmus wie auch der Hug Router. Er verlegt max. im 1 mil.-Raster, also praktisch rasterlos und schmiegt die Bahnen aneinander an. Im zweiten Lauf wird ein Rerante mit weiteren Steuerparametern versucht. Im dritten Lauf (Cleanup-Pass) minimiert er die Vias, begradigt die Bahnen und entfernt Hangers.

Eine praktikable Möglichkeit ist die Vergabe von Routing-Prioritäten. So können breitere Bahnen zuerst, und in einer absteigenden Reihenfolge nach Prioritätsliste (Breitenparameter aus der Netzliste) die schmaleren Bahnen geroutet werden. Der Router berücksichtigt hierbei die Werte aus den Informationsfeldern der Symbole im Schaltbild. Gleichermaßen verwendet er die Via-Information. Ergänzend kann ein Stop in Abhängigkeit von der Breiten-Vorgabe aus der Prioritätenliste vorgegeben werden. Man kann das Ergebnis bis zu diesem Stand ansehen oder auch interaktiv weiterrouten. Ein weiterer Parameter bestimmt den Kompromiß zwischen minimaler Rechenzeit und maximalem Auflösungsgrad.

Der CAM-Output

Gerber: Bis zu 100 Aperture-Codes werden vom System verwaltet. So ist eine Anpassung an verschiedene Blendenteller gegeben. Für die Ausgabe auf Laser-Photoplotter sind dem Anwender durch die hohe Zahl der möglichen Apertures weitere Freiheiten gegeben.

Excellon: Die Bohrdaten werden im Excellon Format ausgegeben.

Plotter/Printer: Die Grafiken können auch auf einem Matrixdrucker ausgegeben werden. Voreinstellung ist hier das Epson-Format mit 120 oder 240 dpi. Momentan wird noch auf DIN A4 ausgegeben. Auf NEC-Druckern kann über HP-GL aber bis zur maximalen Druckformatbreite (DIN A3) ausgegeben werden.

Der Benutzer kann über einen Parameter die Bildschirmdarstellung auf dem Zeichengerät skalieren. So kann auch in metrischem Format ausgegeben werden. Der Auflösungsgrad von Kreisen ist durch einen Parameter steuerbar. Bei einer Layoutausgabe und runden Pads wird bei Setzen eines geringeren Auflösungsfaktors eine deutlich höhere Geschwindigkeit erzielt. Unterstützt werden die HP-Plotter HP7475, HP7580,85,86. Für andere Ausgabegeräte kann eine Initialisierungsdatei erstellt werden. Bei Penplotausgaben kann für einen höheren Schwärzungsgrad ein Überlappungsfaktor gewählt werden. Hier muß der Anwender zwischen diesem Parameter und den anderen Einflußfaktoren wie zum Beispiel der Tuschart, der Folie, des Stifts, der Auflagekraft, der Plotgeschwindigkeit experimentieren. Der Anwender kann auch andere als im Plotter vorhandene Stifte benutzen. Ein Parameter erlaubt eine Unterbrechung bei Stiftwechsel.
Ein wichtiger Punkt für die Ausgabe von Layouts auf Plottern ist die Möglichkeit, die Plottergenauigkeit zu erhöhen. Dies erfolgt durch Setzen eines Parameterwertes jeweils für die X- und Y-Koordinate. Damit werden mechanische Ungenauigkeiten minimiert. Die Erstellung der einzelnen Artwork-Dateien (einzelne Cu-Lagen, Bohrplan, Lötmaske usw.) werden ebenfalls durch BatchDateien unterstützt.
Über das EDIF-Format, aus Future Net, Orcad, HP, Racal-Redac und Scicards können Netzlisten übernommen werden. □

TOSHIBA: Der Spezialist für Portable Computer.

Leichtathlet.

**Wer überall alles wissen muß, ist auf tragbare Computer angewiesen.
Portable Computern gehört die Zukunft.
TOSHIBA will diesen technologischen Vorsprung mit Portable Computern allen schon heute zur Verfügung stellen.**

AT-kompatibel, mit Plasma-Bildschirm und 20 MB Festplatte, der Leichtathlet von TOSHIBA – T 3100.

BERATUNGSGUTSCHEIN

T 1000

T 1100 Plus

T 1200

T 3100

T 3200

T 5100

Hier sehen Sie alle TOSHIBA Portable Computer auf einen Blick. Um genau zu wissen, welcher der Richtige ist, sollten Sie sich beraten lassen.
Wenn Sie Informationsmaterial über die Portables von TOSHIBA haben wollen, schicken Sie den ausgefüllten Beratungsgutschein bitte an TOSHIBA.
Oder wenden Sie sich direkt an einen AUTORISIERTEN TOSHIBA FACHHÄNDLER, dort können Sie die Portables auch im Original sehen.

Firma/Branche

Name

Straße

PLZ Ort

Telefon

MC 988/31

In Touch with Tomorrow
TOSHIBA

KOPIERER • COMPUTER • DRUCKER • FERNKOPIERER

TOSHIBA Europa (I.E.) GmbH, Bereich Deutschland, Abteilung Marketing IPS, Hammer Landstraße 115, 4040 Neuss.

Günther Sternberg

Digitalisier-Tabletts aller Größen

Trotz der erheblichen Preisdifferenz zwischen Maus und Digitalisier-Tablett – ein brauchbares Tablett ist unter 1000 DM fast nicht erhältlich – steigt die Nachfrage besonders im CAD-Bereich sprunghaft an. Wir haben fünf Digitalisier-Tabletts von Graphtec unter die Lupe genommen.

Nachdem in mc bisher nur wenig über Digitalisier-Tabletts zu lesen war und wohl nicht jeder Leser mit ihrer Funktionsweise vertraut ist, zunächst eine kleine Einführung.
Die getesteten Digitalisier-Tabletts arbeiten alle nach dem Prinzip der elektromagnetischen Induktion. Unterhalb der für die Digitalisierung zur Verfügung stehenden Fläche ist eine XY-Matrix von Drähten eingelassen. Die Lupe oder der Stift, das Instrument in der Hand des Benutzers zur Fixierung einer bestimmten Position, besteht in der Hauptsache aus einer Spule. Schickt man nun durch einen Draht der XY-Matrix einen Stromimpuls, so baut sich um den entsprechenden Draht kurzzeitig ein Magnetfeld auf. Nach den Gesetzen der Physik führt ein Wechsel der magnetischen Feldstärke innerhalb einer Spule zur Induktion einer Spannung. Die Größe der induzierten Spannung ist von der Lage der Aufnehmerspule zum entsprechenden Draht abhängig. Zur Bestimmung der Lupenposition muß man also nur durch jeden Draht der XY-Matrix einen kurzen Stromstoß hindurchschicken und die Größe der dadurch in der Lupe induzierten Spannung mit einem AD-Wandler messen. Die größte gemessene Spannung in X- und Y-Richtung läßt sich je einem Draht zuordnen, dessen Lage die X- bzw. Y-Koordinate des gesuchten Punktes ergibt. Das so ermittelte Koordinatenpaar wird über die serielle Schnittstelle an den angeschlossenen Computer übertragen. Natürlich ist dies eine stark vereinfachte Schilderung der Arbeitsweise eines Digitizers.
Aus dem genannten Funktionsprinzip elektromagnetischer Digitalisier-Tabletts läßt sich eine wichtige Verhaltensregel ableiten. Wegen des ständig auf der Oberfläche eines eingeschalteten Digitizers vorhandenen Magnetfelds haben Disketten auf ihm nichts verloren. Zwar sind die Magnetfelder moderner Digitizer nicht so stark, daß eine Diskette sofort gelöscht würde, aber um es mit einem Sprichwort zu sagen, steter Tropfen höhlt den Stein.

Gute Erfahrungen mit Digitalisier-Tabletts

Der japanische Hersteller Graphtec, mit deutscher Vertretung in Herrsching am Ammersee, ist in der Branche vor allem als Lieferant hochwertiger Plotter bekannt. Die Käufer der größeren Griprollen-Plotter (bis Format A0) sind hauptsächlich im CAD-Bereich tätig. Da bei CAD-Anlagen das Digitalisier-Tablett schon seit langem als das wichtigste Eingabemedium gilt, beschloß Graphtec schon vor vielen Jahren, sein Lieferprogramm mit Digitizern aus eigener Entwicklung und Produktion abzurunden. Die neue Baureihe löst die bisherigen Modelle ab und enthält dank der bisherigen Erfahrungen einige Verbesserungen.
Sie besteht aus fünf Modellen, die sich in der Größe der effektiven Digitalisierfläche (*Bild 1*) unterscheiden. Die Flächenmaße der Geräte sind:

KD3200 297 × 210 mm (DIN A4)
KD3300 305 × 305 mm (12" × 12")
KD3800 380 × 380 mm (15" × 15")
KD4300 380 × 260 mm (15" × 10")
KD4600 460 × 310 mm (DIN A3)

Beim Design der neuen Modelle wurden wichtige Punkte der Ergonometrie berücksichtigt. So ist das KD4300, als Stellvertreter der neuen Serie, wesentlich flacher als sein doch recht klobig wirkender Vorgänger, das KD4030. Die geringe Höhe von 2,5 cm an der Vorkante des Gerätes und die Neigung der Arbeitsfläche von 4,5° sorgen für ein angenehmes Arbeitsgefühl.
Die Form der Lupe erscheint auf den ersten Blick etwas futuristisch, gehört aber wohl mit zu den am besten durchdachtesten Teilen des Geräts. Dank der ungewöhnlichen Form liegt die Lupe gut in der Hand und man hat trotz natürlicher Armstellung immer einen guten Blick auf das Fadenkreuz. Einziger Kritikpunkt ist der stellenweise auf der Unterseite der Lupe angebrachte Filz. Er ist bei den beiden Testgeräten et-

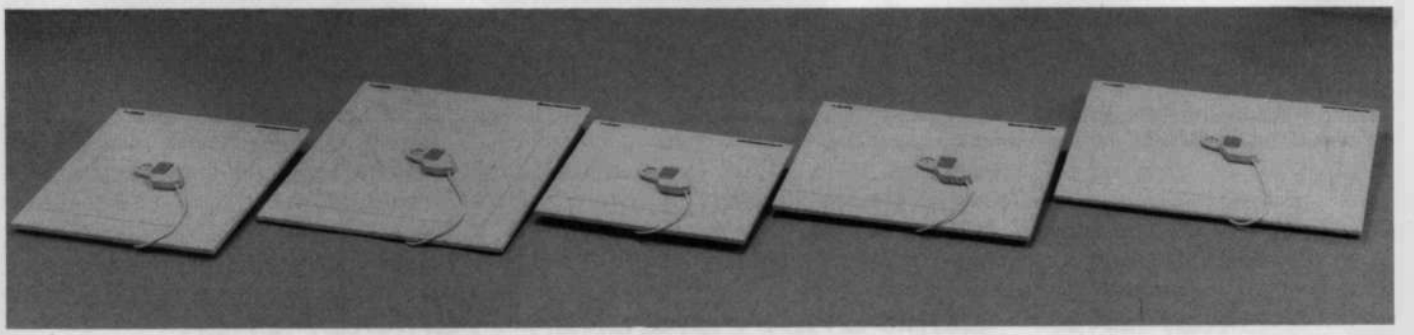


Bild 1. Die neue Serie umfaßt Tabletts in allen gängigen Größen

was zu dünn ausgefallen, wodurch die Lupe leicht auf der Arbeitsoberfläche kratzt.
Auch für das einem meist im Weg liegende Kabel der Lupe wurde eine Lösung gefunden. Die Lupe wird an der Rückseite des Tabletts angeschlossen, das Kabel aber an der Unterseite des Gerätes in einem dafür vorgesehenen Schlitz zur Mitte der Vorderkante geführt. Damit ist das Kabel so wenig wie möglich im Wege. Apropos Kabel, das Netzkabel wird über einen Kaltgerätestekker angeschlossen.
Am rechten oberen Rand des Tabletts ist eine Leiste mit drei Leuchtdioden angebracht. Daneben befinden sich die Aufschriften „Status", „Ready" und „Power". Die rote Leuchtdiode „Status" erfüllt zwei Aufgaben. Zum einen läßt sie sich vom Computer aus per Befehl ein- und ausschalten und zeigt somit zum Beispiel die Eingabebereitschaft des Programms an. Zweitens zeigt sie an, daß sich die Lupe außerhalb oder in zu großer Höhe des Digitalisierbereichs befindet. Die „Ready"-Leuchtdiode zeigt die Sendebereitschaft und damit den Zustand der Handshake-Leitungen der seriellen Schnittstelle an.
Der Einschalter des Netzgerätes, der Reset-Knopf und die drei DIP-Schalter zur Einstellung der Gerätekonfiguration befinden sich, sehr gut erreichbar, auf der rechten Seite des Tabletts unterhalb der Arbeitsfläche. Lobenswert ist die Tabelle mit den Erklärungen zu den verschiedenen Einstellungen der DIP-Schalter auf der Unterseite des Gerätes. Damit erspart man sich häufig den Griff zum Handbuch und die Information kann nicht verlorengehen.

Software-Anpassung erforderlich

Der Lieferumfang umfaßt neben dem Tablett die Lupe mit vier Tasten (Stift mit einer Taste und Lupe mit sechs Tasten als Option), ein knapp 100seitiges englisches Handbuch, eine Staubhülle, das Netzkabel und eine Ersatzsicherung.
Das zum Anschluß des Tabletts benötigte serielle Schnittstellenkabel hängt vom jeweiligen Computertyp ab. Wegen der zu großen Anzahl unterschiedlicher Kabel liefert Graphtec kein Kabel mit. Graphtec ist jedoch in der Lage, für die gängigsten Computertypen das entsprechende Kabel zu liefern.
Ein Kapitel des ausführlichen Handbuchs widmet sich dem Test der Verbindung zwischen Tablett und Computer. Für jeden der folgenden Computer ist ein kleines Programm (meistens in Basic) zum Test der Verbindung vorhanden, und alle für diesen Test nötigen hard- und softwaremäßigen Einstellungen sind beschrieben:

Tabelle 1: Datenformate der Graphtec-Digitizer

ASCII-Format

±XXXX,±YYYY,F CR LF

±	Vorzeichen beim Relativ-Modus
XXXX	horizontale Lupenposition als ASCII-Zahl
YYYY	vertikale Lupenposition als ASCII-Zahl
F	Code für gedrückte Lupentaste(n) als ASCII-Zahl 1 = "Z"-Taste gedrückt 2 = Taste 1 gedrückt 4 = Taste 2 gedrückt 8 = Taste 3 gedrückt
CR	ASCII-Code (0DH) für Carriage-Return
LF	ASCII-Code (0AH) für Linefeed

5-Byte-Binär-Format

Bit	7	6	5	4	3	2	1	0
Byte 1	0	1	F3	F2	F1	F0	XA	YA
Byte 2	0	0	X5	X4	X3	X2	X1	X0
Byte 3	0	0	X11	X10	X9	X8	X7	X6
Byte 4	0	0	Y5	Y4	Y3	Y2	Y1	Y0
Byte 5	0	0	Y11	Y10	Y9	Y8	Y7	Y6

X11-X0	horizontale Lupenposition
Y11-Y0	vertikale Lupenposition
F0-F3	Zustand der Lupentasten F0 = 1 → "Z"-Taste gedrückt F1 = 1 → Taste 1 gedrückt F2 = 1 → Taste 2 gedrückt F3 = 1 → Taste 3 gedrückt
XA,YA	Vorzeichen beim Relativ-Modus XA = 1 → negative X-Koordinate YA = 1 → negative Y-Koordinate

8-Byte-Binär-Format

Bit	7	6	5	4	3	2	1	0
Byte 1	0	1	0	0	0	0	1	PR
Byte 2	0	0	0	0	F3	F2	F1	F0
Byte 3	0	0	X5	X4	X3	X2	X1	X0
Byte 4	0	0	X11	X10	X9	X8	X7	X6
Byte 5	0	0	0	XA	X15	X14	X13	X12
Byte 6	0	0	Y5	Y4	Y3	Y2	Y1	Y0
Byte 7	0	0	Y11	Y10	Y9	Y8	Y7	Y6
Byte 8	0	0	0	YA	Y15	Y14	Y13	Y12

PR	Zustand der Lupe PR = 1 → Lupe außerhalb der Digitalisierfläche
X15-X0	horizontale Lupenposition
Y15-Y0	vertikale Lupenposition
F0-F3	Zustand der Lupentasten F0 = 1 → "Z"-Taste gedrückt F1 = 1 → Taste 1 gedrückt F2 = 1 → Taste 2 gedrückt F3 = 1 → Taste 3 gedrückt
XA,YA	Vorzeichen beim Relativ-Modus XA = 1 → negative X-Koordinate YA = 1 → negative Y-Koordinate

- Apple II
- Hewlett-Packard HP85 und HP9816A
- IBM PC/XT/AT
- NEC PC-9801/E/F und PC-8801/mkII
- NEC PC100
- NEC PC-8001mkII
- Fujitsu FM-7/New 7, FM-8, F-77, FM-11
- Oki iF800 Model 20/30/50

Nachdem man sich des prinzipiellen Zusammenspiels von Tablett und Computer vergewissert hat, kann man sich der Konfiguration des Tabletts in Hinblick auf eine bestimmte Software zuwenden.
Als erstes gilt es, die serielle Schnittstelle (RS-232C) des Tabletts auf die entsprechenden Angaben des Software-Herstellers ein-

Tabelle 2: Alle wichtigen Funktionen des Tabletts lassen sich per Befehl ändern

T	Test-Modus – Bis zum Ausschalten des Digitizers werden alle empfangenen Zeichen an den Computer zurückgesendet	S	Output Inhibit – Sperren der Datenübertragung
		Y	Beim Empfang dieses Befehls sendet der Digitizer die aktuelle Lupenposition und den Tastenzustand an den Computer
U	Status-Lampe ein		
V	Status-Lampe aus	a	Relativ-Modus aktivieren – Die Lupenposition wird ab jetzt relativ zur jeweils letzten übertragenen Lupenposition berechnet
Z	Signalgeber des Digitizers für 40 ms aktivieren		
Wd	Mindestabstand zwischen zwei auszugebenden Punkten im Stream- und Switch-Stream-Modus festlegen	b	Relativ-Modus ausschalten
		c x,y CR LF	Der Koordinaten-Ursprung wird an die Position X/Y des Digitizers verschoben
R	Mindestabstand aufheben		
P	Point-Modus aktivieren – Lupenposition und Tastenzustand wird bei jeder Betätigung einer Lupentaste übertragen	c	Der Koordinaten-Ursprung wird wieder in die linke untere Ecke des Digitizers gelegt
		e x0,y0,x1,y1 CR LF	Winkelkompensation ein – Anhand der beiden Koordinatenpaare X0/Y0 und X1/Y1 wird der Drehwinkel des Koordinatensystems berechnet. Alle zukünftigen Koordinaten werden um diesen Drehwinkel korrigiert
H bis O	Stream-Modus aktivieren – Je nach Kennbuchstabe werden 2 bis 60 Meßwerte pro Sekunde übertragen		
@ bis G	Switch-Stream-Modus aktivieren – Je nach Kennbuchstabe werden 2 bis 150 Meßwerte pro Sekunde übertragen	f	Winkelkompensation aus

zustellen. Erlaubt sind Übertragungsraten zwischen 150 und 9600 Baud, 7 oder 8 Datenbits, 1 oder 2 Stopbits und die Wahl zwischen keiner, gerader oder ungerader Parität. Für die Übertragung von Position und Tastenzustand der Lupe gibt es die Wahl zwischen drei Datenformaten (*Tabelle 1*). Das 5-Byte-Binärformat gehört zu den am häufigsten verwendeten Datenformaten und wird von den meisten Programmen benutzt.

Der nächste hardwaremäßig, also per DIP-Schalter, festzulegende Punkt ist die Wahl der Betriebsart, die beim Einschalten des Digitizers bis zu einer eventuellen Änderung per Software aktiv sein soll. Zur Wahl steht der Point-, Remote-, Stream- oder Switch-Stream-Modus. Beim Point-Modus wird die Position der Lupe und der Tastenzustand jeweils bei Betätigung einer der Lupentasten übertragen, wobei im Remote-Modus diese Daten ausschließlich nach Anforderung durch den Computer übermittelt werden. Im Stream-Modus sendet das Tablett die Positionsangabe und den Tastenzustand ununterbrochen an den Computer. Die Zahl der pro Sekunde zu übertragenden Meßwerte wird ebenfalls per DIP-Schalter festgelegt. Der Switch-Stream-Modus unterscheidet sich vom Stream-Modus nur darin, daß die kontinuierliche Datenübertragung nur während der Betätigung einer der Lupentasten erfolgt.

Sollte sich das verwendete Software-Paket aber mit keinem der drei Graphtec-Formate zufrieden geben oder einfach kein Treiber für einen Graphtec-Digitizer vorgesehen sein, ist längst nicht alles verloren. Die Digitizer der neuen Serie sind in der Lage, die Tabletts der HDG-Serie von Hitachi, das Tablett DT 4100 von Seiko und die Tabletts der „Bit-Pad-Two"-Baureihe und der MM-Serie von Summagraphics weitgehendst hard- und softwaremäßig zu emulieren. Damit lassen sich wohl die meisten Programme mit den Graphtec-Digitizern zum Laufen bringen.

Eine Besonderheit der Digitizer gilt es noch zu erwähnen. Mit Hilfe eines zusätzlichen EPROMs sind alle Digitizer dieser Serie in der Lage, das Übertragungsformat einer Microsoft-Maus zu erzeugen. Zusammen mit einem Maustreiber entsprechend dem Microsoft-Standard könnte das Tablett als Mausersatz fungieren. Leider ist derzeit ein solcher Treiber nicht verfügbar.

Intelligentes Digitalisier-Tablett

Digitizer werden meist speziell für eine Aufgabe, sprich zu einem bestimmten Programm, angeschafft. Hat man sich erst einmal an die Vorteile des Gerätes gewohnt, so reift rasch der Wunsch, auch andere Aufgaben damit zu lösen, also unter Umständen den Digitizer von eigenen Programmen heraus anzusteuern. Graphtec-Digitizer unterstützen diesen Wunsch durch eine Reihe von Kommandos mit einer gewissen Eigenintelligenz.

Jedes Kommando besteht aus einem Kennbuchstaben, dem bei Bedarf die entsprechenden Parameter folgen. *Tabelle 2* faßt die Befehle zusammen. Besonders hervorzuheben sind meines Erachtens vier Befehle. So kann man per Befehl den Ursprung des Koordinatensystems an eine beliebige Stelle der Digitalisierfläche verschieben. Auch der Wechsel von absoluten zu relativen Koordinaten ist vorgesehen.

Eine wichtige Arbeitserleichterung bei der Digitalisierung von Kurven bildet der Inkrement-Modus. Damit läßt sich beim Stream- und Switch-Stream-Modus ein Mindestabstand zwischen zwei zu übertragenden Positionen festlegen. Möchte man also eine beliebige Kurve durch Punkte im Abstand von 2 mm digitalisieren, muß man lediglich den Mindestabstand per Befehl festlegen und dann die Kurve mit der Lupe abfahren. Alles andere erledigt die Software des Digitizers. Ebenfalls äußerst nützlich ist die Drehung des Koordinatensystems. Es ist sehr mühsam, ein Blatt Papier mit einer zu digitalisierenden Kurve parallel zu den Achsen des Digitizers auszurichten. Dank der Winkelkompensation legt man das Papier einfach beliebig auf das Tablett und digitalisiert zwei Punkte auf der X-Achse des Blattes. Die Koordinaten dieser beiden Punkte werden dann als Parameter beim „e"-Befehl angegeben. Damit wird das Koordinatensystem des Digitizers um den aus den beiden Punkten berechneten Winkel gedreht und alle weiteren Meßpunkte entsprechen den Werten eines ideal ausgerichteten Blattes.

Positives Gesamturteil

Neben dem ergonomischen Design von Tablett und Lupe gilt es vor allem, die Eigenintelligenz und die Vielzahl an Betriebsarten lobend hervorzuheben. Das deutsche Handbuch befindet sich bereits in Vorbereitung und wird in Zukunft hoffentlich kostenlos beigelegt. Schließlich zählt Graphtec aufgrund der Präzision der Digitizer nicht zu den Low-Cost-Anbietern, so daß eine deutsche Dokumentation im Preis inbegriffen sein sollte. □

Horst Brand

Titan sucht mit Erfolg

Ein Retrievalsystem im Test

Volltextdatenbanken, oft auch Retrievalsysteme genannt, eignen sich zum schnellen, unkomplizierten Auffinden bestimmter Daten oder Texte innerhalb großer Datenbestände. Im Unterschied zu relationalen Datenbanksystemen kann das Material auch völlig unstrukturiert vorliegen.

Das Hauptmenü nach dem Start von Titan (*Bild*) besteht aus einer bildschirmfüllenden Blockgrafik. Man muß dort ein einzelnes Angebot mit den Cursortasten ansteuern, bevor man eines mit der Enter-Taste auswählen kann. Eine Beantwortung von Menüs mit Shortcuts (Eingabe nur von Anfangsbuchstaben) ist im gesamten Programm nicht vorgesehen; das vermißt man spätestens dann, wenn man nach der Einarbeitungsphase möglichst rasch ans Ziel kommen will.

Am schnellsten kennenlernen lassen sich die wesentlichen Eigenschaften von Titan, wenn man die Schnittstelle Textverarbeitung benutzt. Neben dem ASCII-Format weist das Hauptmenü aus, daß es mit folgenden speziellen Text-Formaten fertig wird: Tex Ass, Wordstar 4.0, Wordstar 2000, Word-Perfect und MS-Word. Zur Erfassung bereit steht jeweils das momentane MS-DOS-Directory, das mit F1 gewechselt werden kann. Alle Dateien eines Verzeichnisses sind aufgeführt und können wiederum mit den Cursortasten angesteuert und ausgewählt werden.

Aus dem Zusammenhang gerissen

Eine solche einzelne Datei wird in Titan unter der Bezeichnung „Titel" geführt. Zunächst gibt man dem Titel eine „Überschrift". Zur Identifikation trägt außerdem ein Datum bei, das mit dem aktuellen Datum vorbelegt ist, aber geändert werden kann. Jetzt kann man den Titel komplett mit Berücksichtigung jedes einzelnes Begriffes (ausgenommen sind nur die Wörter der Stopwortliste) in die Indexdatei übernehmen lassen. Entscheidet man sich gegen eine Erfassung in einem Zuge, dann wird der Titel bildschirmseitenweise angezeigt; es sind nun jeweils Änderungen erlaubt (kleine Korrekturen oder auch Löschen von ganzen Zeilen) oder es kann die ganze Seite herausgenommen werden. Damit ist der Titel zunächst nur zwischengespeichert. Zur vollen Integration in das Retrievalsystem mit Berücksichtigung seiner Begriffe in der Indexdatei muß ausdrücklich ein sogenannter Reorganisationslauf durchgeführt werden.

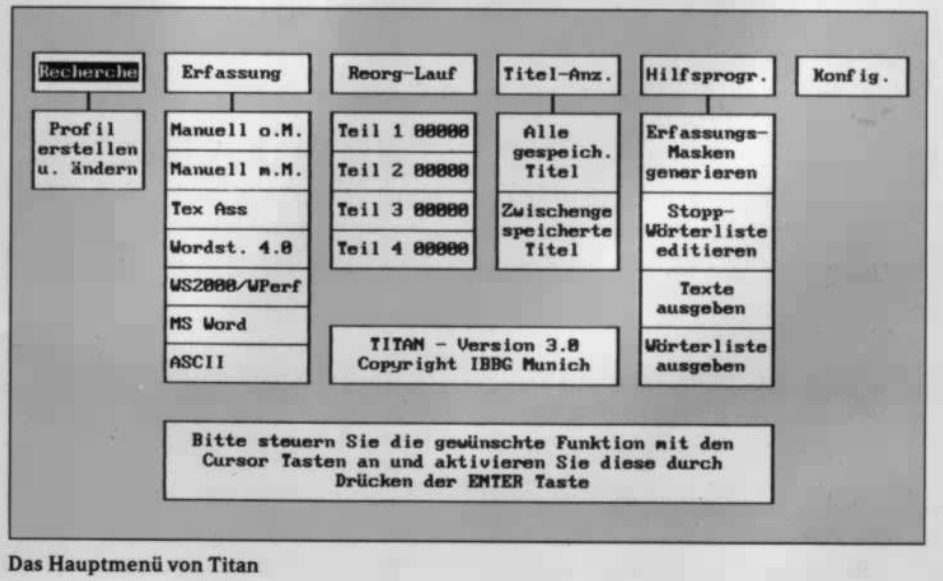
Zur manuellen Eingabe von Texten stehen die elementaren Textverarbeitungsfunktionen zur Verfügung. Dabei kann man auch mit Masken arbeiten, was jedoch nur bedeutet, daß die neue Bildschirmseite mit dem Maskentext vorbelegt wird, den man nun überschreiben oder auch beibehalten kann.

Titan speichert alle Texte in einer einzigen großen Datei mit dem Namen DOD.DAT. Wenn man die maximale Kapazität von 16 MByte ausschöpft, belegt das System (zusammen mit Programm, Indexdatei und Hilfsdateien) insgesamt ca. 32 MByte.

Die maßgebliche Texteinheit ist eine Bildschirmseite. Unter dem Menüpunkt „Titel-Anzeige" werden in einer Übersicht alle Seiten aufgeführt, zusammen mit Überschrift und Datum des Titels, zu dem sie gehören. Jede Seite kann nach der Anwahl in der Übersicht einzeln angeschaut werden; „Blättern" (mit einer Taste) ist unter dem Menüpunkt „Titel-Anzeige" nicht möglich.

Nachträgliche Korrekturen können ebenfalls unter dem Menüpunkt „Titel-Anzeige" durchgeführt werden. Bei den Recherchen (siehe unten) kann allerdings mit einer Taste geblättert werden, Korrekturen sind dort ebenfalls ausführbar. Unverständlicherweise werden nachträglich geänderte Seiten aus dem Zusammenhang gerissen und an das Ende der Titan-Datei gesetzt. Das kann ein Blättern innerhalb eines Titels in der Umgebung einer Fundstelle sinnlos machen. Ohne nachträgliche Korrekturen bleibt die Anordnung der Titel in der Reihenfolge der Erfassung erhalten. Die angebotenen Sortiervorgänge nach Datum oder Titelüberschrift wirken sich nur auf die Titelübersicht aus und sortieren für jede Bildschirmseite der Titelübersicht getrennt (Screen-Sort).

Den eigentlichen Nutzen eines Retrievalsystems für den Anwender, nämlich das schnelle Auffinden von bestimmten Textstellen, bietet Titan mit den sogenannten Recherchen. Bei der einfachsten Form der Recherche wird nur ein Suchwort eingegeben; Titan gibt die Zahl der Treffer an, PgUp und PgDn zeigen die betroffenen Textseiten in aufsteigender bzw. absteigender Reihenfolge an. In der Testversion von Titan, die der Redaktion zur Verfügung



Das Hauptmenü von Titan

Zwischen Textverarbeitung und relationaler Datenbank

Suchen gehört zu den elementaren Aufgaben der Datenverarbeitung. Je größer eine Informationsmenge wird, um so leichter verliert der Benutzer die Übersicht und umso wichtiger ist es, daß mächtige, aber einfach zu handhabende Werkzeuge zum Wiederauffinden zur Verfügung stehen.

Umgekehrt kann man sagen: solange ein Bestand noch überschaubar und dem Benutzer in allen Einzelheiten jederzeit präsent ist, kann Datenverarbeitung zur besseren Bewältigung eigentlich nichts beitragen. Die Zeiten, in denen man den Heimcomputer auf den 100-Flaschen-Weinkeller ansetzte, scheinen aber nun endgültig vorbei.

Serielle Suchfunktionen gehören zu fast jedem Anwenderprogramm, jeder bessere Editor (zum Beispiel von Programmiersprachen) hat sie. Dabei wird ein Text solange untersucht, bis eine bestimmte Zeichenfolge (String) gefunden ist. Wegen der simplen Methode ist eine serielle Suchfunktion selbst schon recht langsam; wenn mehrere Dateien als „Geheimnisträger" in Frage kommen, muß unter Umständen in jeder einzelnen gesucht werden.

Suchvorgänge komplexerer Natur sind die Anfragen an ein relationales Datenbanksystem: Datensätze werden nach bestimmten Kriterien gefiltert und aussortiert. Bei einer Bestell-Liste könnte das in einem Extremfall so aussehen: Gesucht sind alle Bestellungen, die von Leuten aus der Heimatstadt von Herrn Schmidt in den letzten 90 Tagen vorgenommen wurden, aber mit den vorhandenen Lagerbeständen nicht mehr ausgeführt werden können. Es sollen der Wert der Bestellung und die jeweils fehlende Stückzahl mitgeteilt werden, zu berücksichtigen sind dabei lediglich Artikel aus Holz.

Der Einsatz eines relationalen Datenbanksystems ist dort sinnvoll, wo Datenmaterial sich leicht strukturieren läßt oder bereits strukturiert vorliegt. Für eine Adreßverwaltung zum Beispiel liegt es nahe, einen Datensatz in die Felder Name, Straße mit Nummer, Postleitzahl, Stadt, Land und Telefonnummer zu unterteilen. Doch daß die Struktur mit Bedacht gewählt sein will, zeigen zum Beispiel folgende Fragen.

Sollten zum Beispiel Nachname und Vorname jeweils ein eigenes Feld einnehmen?

Weil man eine Datei nach jedem einzelnen Feld (alphabetisch bzw. nach Größe der Zahlen) sortieren kann, es aber (es sei denn innerhalb eines Freundeskreises) keinen großen Sinn macht, nach Vornamen zu sortieren, kommt man mit nur einem Namensfeld aus. Dort muß aber der Nachname vorangestellt werden, damit er beim Sortieren Vorrang hat.

Kann man Postleitzahl und Ort nicht zu einem Feld zusammenfassen?

Nach Städten zu sortieren, kann zweckmäßig sein. Aber auch Sortieren nach Postleitzahlen kann nützlich sein (zum Beispiel für gezieltes Anschreiben von Adressen innerhalb bestimmter Gebiete). Beide sollten also in getrennten Feldern bleiben.

Derartige Überlegungen sind bei relationalen Datenbanken an der Tagesordung. Je größer die Zahl der Felder ist, umso mehr muß man sich über die Verteilung der Felder auf mehrere Tabellen Gedanken machen, auch um Redundanz zu vermeiden und Speicherplatz zu sparen. Das macht die Struktur komplizierter und schafft das Problem, die Tabellen bei bestimmten Anfragen (je nach Datenbanksystem mehr oder weniger umständlich) wieder verknüpfen zu müssen.

Schnelles Suchen auch innerhalb sehr großer Datenbestände wird in relationalen Datenbanksystemen durch die sogenannte Indexierung ermöglicht. Dabei wird für jedes indexierte Feld eine eigene Datei verwaltet, in der die Elemente ständig in der richtigen (absteigenden oder aufsteigenden) Reihenfolge gehalten werden und sich außerdem nur noch die zugehörigen Nummern der Datensätze befinden. Diese Indexdateien werden bei jeder Änderung der Hauptdatei sofort aktualisiert. Mit dem sogenannten binären Suchverfahren reichen dann zum Beispiel schon 20 Zugriffe (bestehend im Wesentlichen jeweils nur aus einer Größer-oder-kleiner-Abfrage) aus, um aus etwas über 1 Million Datensätze den richtigen herauszufinden.

Fassen wir zusammen: der unbestreitbare Vorteil von relationalen Datenbanken liegt in ihren vielfältigen Sortier- und Selektionsmöglichkeiten, die sehr schnell ausgeführt werden können. Hinzu kommen freie Gestaltungsmöglichkeiten für Masken zur Eingabe neuer Datensätze und für Listen selektierter Datensätze, die weitgehend losgelöst von der internen Struktur aufgebaut werden können. Um in den Genuß dieser Flexibilität zu kommen, muß aber erst einmal eine sinnvolle Struktur her und das Material (falls überhaupt möglich) in diese Struktur gepreßt werden. Das ist ihr Nachteil. Einige relationale Datenbanksysteme tun sich zudem schwer in der Umstrukturierung von Dateien.

Ganz anders die Retrievalsysteme. Sie verlangen keine Struktur. Bei einigen von ihnen wird jeder einzelne vorkommende Begriff (Wort oder Zahl) automatisch in eine Indexdatei geschrieben. Das macht Suchvorgänge auch in sehr großen Dateien noch schnell. Von der Aufnahme in die Indexdatei kann man Wörter mit der sogenannten Stopwortliste ausnehmen. „Der", „in", „auf" sind zum Beispiel solche Wörter, nach denen man wohl niemals suchen wird.

Besonders geeignet sind Retrievalsysteme für große Textmengen (Archive, Dokumente, Bücher, Zeitschriften); deswegen werden sie auch Volltext-Datenbanken genannt. Aber auch Anwendungen, die wegen der Strukturiertheit der Daten eigentlich typisch für ein relationales Datenbanksystem sind, lassen sich mit einem Retrievalsystem verwirklichen, zum Beispiel die oben bereits erwähnte Adreßverwaltung.

Auch bei vielen Retrievalsystemen sind Einteilungen in Felder möglich, auf die man Suchvorgänge beschränken kann. Erforderlich sind sie aber meistens nicht. Um zum Beispiel die Telefonnummer von Herrn Huber zu erfahren, genügt es, den Begriff „Huber" in einer unstrukturierten Datei zu suchen und seine Umgebung anzuzeigen. Verwechslungen mit anderen Feldern sind (zumindest bei einer Adreßverwaltung) selten; daß auf der Suche nach der Adresse von Frau Stein auch die Adresse von Herrn Schmidt, der in der Stadt Stein wohnt, angegeben wird, kann man in Kauf nehmen. Es kann sogar erwünscht sein, daß zum Beispiel ein Personenname unter der Privatadresse und als Kontaktperson einer Firma gefunden wird.

stand, wurden die Treffer an den Fundstellen zunächst nicht markiert. Mit einer Änderung über den Menüpunkt „Konfiguration" kann man alle gefundenen Textstellen blinken lassen, so daß auf einen Blick der genaue Ort und die Anzahl der Treffer erkennbar sind. Inzwischen wird eine Version ausgeliefert, bei der die Konfiguration schon in dieser Weise voreingestellt ist.
Wenn auf der gleichen Seite sowohl das Suchwort als auch ein längeres Wort, an dessen Anfang das Suchwort steht, angetroffen werden, dann blinkt auch der betroffene Teil des längeren Wortes auf. Wirklich als Treffer akzeptiert wird es aber nicht, denn die Trefferanzahl erhöht sich nicht und auf anderen Seiten blinkt der betroffene Teil des längeren Wortes nicht. Gar nicht erkannt werden längere Wörter, in denen das Suchwort weiter hinten vorkommt. Konsequenter wäre gewesen, nur das genaue Suchwort aufblinken zu lassen.
Die Eingabe weiterer Suchwörter stellt eine Und-Verknüpfung zwischen ihnen dar. Alle und-verknüpften Suchworte müssen innerhalb einer Seite stehen, damit sie in ihrer Kombination gefunden werden. Es kann also passieren, daß zwei Suchworte im Text unmittelbar aufeinander folgen, sie aber bei Und-Verknüpfung nicht gefunden werden, weil genau zwischen ihnen eine neue Seite anfängt. Oder-Verknüpfung geschieht mit einem O vor weiteren Suchwörtern, der Ausschluß bestimmter Suchwörter kann mit dem Operator N (für „not") durchgeführt werden.

Ist eine Suche erfolglos, wird ein Auschnitt aus der Indexdatei angezeigt, der alphabetisch kurz vor und kurz hinter dem Suchwort steht. Dies kann manchmal als Hinweis auf Abweichungen in der Schreibweise dienen, aber nur wenn sie weit hinten im Suchwort liegen und nicht zum Beispiel bei den Wörtern Foto und Photo.
Wie bei Interpretersprachen die Kommandos sofort ausgeführt, aber nicht gespeichert werden, so sind auch die verwendeten Suchwörter-Kombinationen nach einer Recherche verloren. Durch eine Recherche-Rückblende läßt sich zwar das Protokoll der Suchergebnisse der letzten Recherche zurückholen, aber auch nur, solange man den Rechercheteil von Titan nicht verlassen hat. Längere oder oft verwendete Suchwörter-Kombinationen kann man in sogenannten Profilen zusammenfassen und sichern. Aus der Recherche heraus werden sie wie ein einzelnes Suchwort aufgerufen, zur Unterscheidung ist ihnen ein Stern (*) vorangestellt.
Suchwörter können auch mit Wildcards versehen sein: durch sogenannte Rechts-Trunkation mit einem Bindestrich werden alle Wörter gesucht, die von links her mit dem eingegebenen Suchwort übereinstimmen.
Eine Liste von 48 Wörtern, die wohl in jeder Stopwörterliste vorkommen, ist im Handbuch abgedruckt. Bislang mußte der Anwender sie selbst eintippen, in Zukunft ist sie bei der Auslieferung bereits Bestandteil von Titan.

Titan wird von der Informationsbanken-Betriebs GmbH in München angeboten und kostet in der Einplatzversion ca. 1400 DM und in der Mehrplatzversion (4 bis mehr als 32 Plätze) zwischen ca. 3000 DM und ca. 8000 DM (jeweils inkl. Mehrwertsteuer).

Verbesserte Version in Sicht

Als schwerwiegenden Mangel der Version 3.0 muß (wie bereits erwähnt) genannt werden, daß eine korrigierte Bildschirmseite an das Ende der Gesamtdatei verschoben und damit aus einem eventuell bestehenden Zusammenhang gerissen wird. Weil sich ein Retrievalsystem gerade in der unkomplizierten Übernahme bestehender Textdateien von relationalen Datenbanksystemen abheben kann, vermißt man auch eine Funktion, mit der man alle Dateien eines Directories auf einmal transferieren kann.

In der Version 4.0 von Titan, die zur Orgatechnik im Oktober fertig sein soll, will man diese Defizite beseitigt haben. Ein Dokument kann dann bis zu 12 Bildschirmseiten lang sein, bei Korrekturen wird das gesamte Dokument verschoben, der Zusammenhang innerhalb des Dokumentes bleibt also erhalten, auch zeilenweises Scrollen soll dann möglich sein. Unter anderem ist auch Maussteuerung vorgesehen, und die Editiermöglichkeiten sollen verbessert werden. □

AT- und Modell 30-Einplatinen-Computer von Western Digital

Der kalifornische Halbleiter-Hersteller Western Digital hat mit dem WD286-WDM2 einen AT-Einplatinen-Computer vorgestellt. Dieser basiert auf der CPU 80286. Unter der Bezeichnung WD30-WDM gibt es von Western Digital jetzt auch einen Single-Board-Computer, der kompatibel zum IBM-PS/2 Modell 30 ist.

Der zum IBM-AT kompatible WD286-WDM2 arbeitet mit 12,5 MHz ohne Wartezyklen. In dem Einplatinen-Computer sind ein EGA-Adapter, zwei serielle Schnittstellen, ein Tastatur- und ein Maus-Port enthalten. Weiterhin sind ein Reset-, ein Keyswitch-Port und vier Erweiterungs-Slots integriert. Zusätzlich ist noch ein 16-Bit-Disk-Interface-Connector von Integrated Drive Electronics (IDE) und ein spezieller Single-Chip-Floppy-Controller enthalten.

Der WD286-WDM2 koppelt den AT-Bus von der CPU ab und kann so auch bei 6 MHz betrieben werden.

Dadurch ist er kompatibel zu Add-On-Cards von Dritt-Herstellern, die bei einer Taktfrequenz von 8 MHz arbeiten.

Der EGA-Adapter des Einplatinen-AT basiert auf dem PEGA-Video-Controller von Paradise. Dieser Chip unterstützt eine Auflösung von bis zu 640 x 480 und 16 Farben. Außerdem kann er Farben durch 16 Graustufen auf einem Monochrom-Bildschirm simulieren und ist kompatibel zu Hercules- und Plantronics-Colorplus-Grafikkarten.

Der zum Modell 30 der PS/2-Familie kompatible WD30-WDM setzt eine CPU 8086 mit 10 MHz Taktfrequenz ein.
Dieser Einplatinencomputer enthält den Paradise PGA1 Video-Graphic-Adapter (VGA), Floppy- und Festplatten-Controller von Western Digital, zwei serielle und eine parallele Schnittstelle. Außerdem stehen ein Tastatur-, ein Maus- und ein Front-Panel-Anschluß zur Verfügung. Das Bus-Timing dieser Platine ist kompatibel zum IBM-PS/2-System.
Eine andere Version dieser Karte hat zusätzlich noch zwei IDE-Disk-Interface-Verbindungen.
Alle Bausteine sind über das weitgespannte Distributionsnetz von Western Digital erhältlich.
Paradise Systems ist eine 100%ige Tochter von Western Digital. Western Digital ist im kalifornischen Irvine beheimatet und wurde 1970 als Hersteller spezieller Halbleiter gegründet. □

Nach Unterlagen von Western Digital.

Michael Braun, Jochen Guther

Modula-2 wird erwachsen

Die Modula-2-Entwicklungssysteme von ITC, JPI, Logitech und Stony Brook

Modula-2 ist die jüngste der bekannten Programmiersprachen. Das Buch „Programmieren in Modula-2" von Niklaus Wirth enthält die Sprachdefinition. Es liegt momentan in der dritten Auflage vor, an der wir uns auch bei diesem Vergleichstest von vier interessanten Modula-2-Systemen orientiert haben.

Gute Sprachen kommen aus einem Kopf, behauptete Niklaus Wirth und schuf nach einer für ihn enttäuschenden Mitwirkung im Algol-68-Sprachkomitee ('...1968 wurde das Monster geboren...') jeweils im Alleingang die Programmiersprachen Pascal und später Modula-2. Modula-2 tritt wie ADA mit dem Anspruch an, eine universell verwendbare Sprache zu sein, mit der sich Betriebssysteme und Echtzeitanwendungen ebenso realisieren lassen, wie große kommerzielle Softwareprojekte. Ein guter Modula-2-Compiler kommt mit ca. 5000 Zeilen Sourcecode aus, während ADA-Compiler durch die unübersehbare Fülle der Sprachelemente zwischen 40 000 (Meridian AdaVantage) und 350 000 Zeilen (Alsys) Umfang haben und ein Alptraum für jeden Compilerbauer sind. (Ein guter Modula-2 Compiler hat 2 bis 3 Bugs; wie viele Bugs hat ein guter Ada-Compiler?). Modula-2 ist 'Pascal ohne die Nachteile von Pascal' (Jerry Pournelle in Byte). Die Brauchbarkeit für die Praxis steht und fällt aber mit der Verfügbarkeit fehlerfreier und zuverlässiger Entwicklungssysteme. Bei der Entwicklung verschiedener Applikationen und Tools mit insgesamt etwa 4 MByte Sourcecode haben wir Stärken und Schwächen der verschiedenen Compiler beobachtet. Hier sind die Ergebnisse:

Die vier Entwicklungssysteme

Für eine sinnvolle Arbeit muß ein Modula-2 Entwicklungssystem mindestens folgende Komponenten enthalten:

- Compiler
- Linker
- Maker
- Librarian (Bibliotheksverwaltung)
- Standardbibliotheken
- Quellcode der Bibliotheksmodule

Deshalb haben wir für diesen Test jeweils die Komplettpakete verwendet, nicht die teilweise ebenfalls erhältlichen 'Sparversionen', bei denen Sourcetexte und/oder die Make-Utility fehlen. Die Bezeichnungen, Versionsnummern und Preise sind in *Tabelle 1* zusammengestellt. Wenn der Compiler eines Entwicklungssystems Standard *.OBJ Dateien erzeugt, kann man auf einen speziellen Linker und Librarian verzichten und stattdessen die entsprechenden DOS-Programme verwenden. Wir haben dies beispielsweise bei Stony Brook und Logitech mit dem Microsoft OS/2 Linker und Librarian erfolgreich getestet.

Die Handhabung

Die Entwicklungssysteme von ITC und JPI arbeiten beide unter einer Benutzeroberfläche, die Editor, Maker, Compiler und Linker als Overlays aufruft, ohne dabei auf DOS-Ebene zurückzukehren.

ITC

Das ITC-Entwicklungssystem bietet eine in sich geschlossene Desktop-Oberfläche, von der aus Taschenrechner, Uhr, Editor, DOS Shell, Librarian, Maker, Compiler und Linker mit Hotkeys oder über Pulldown-Menüs gewählt werden können. Bildschirmattribute, Tastenfunktionen und Dialogtexte können mit einem ASCII-Editor definiert und mit dem mitgelieferten Resource-Compiler in einer Resource-Bibliothek abgelegt werden. Jede Modul-Bibliothek kann bis zu 256 Module enthalten. Der Librarian zeigt die Namen der Module in einem Fenster: DEF, IMP, MOD mit dem Quelltext, SYM und LNK mit den Resultaten der Compilierung sind unter ein und demselben Namen zugänglich. Das Format der Libraries hat ITC nicht veröffentlicht, „zerschossene" Libraries können daher nicht mit Tricks oder üblichen Utilities gerettet werden. Beim Editieren geänderte Dateien werden beim Abspeichern hinten an die Library angehängt, die sich dadurch stetig weiter aufbläht und von Zeit zu Zeit durch Aufruf von „Compress" verkleinert werden muß. Das System kann beim Compilieren maximal fünf Libraries öffnen. Der ITC-Editor bietet von den vier getesteten den größten Bedienungskomfort: Bereits beim Aufruf des Editors wird das Gerüst eines DEF, IMP oder MOD-Moduls vorgegeben. Für jedes Modula-2-Konstrukt gibt es einen Hotkey, der das vollständige, auch mehrzeilige Konstrukt einschließlich Einrückung und Begrenzer vorgibt; mit der TAB-Taste kann der Programmierer nur die Stellen zum Editieren erreichen, die individuell ausgefüllt werden müssen. So kann man völlig frei von jeder Kenntnis der Modula-2-Syntax fehlerfrei programmieren. Die Cursortasten scrollen nicht nur zeilenweise durch den Quelltext, sondern auch konstruktweise: durch Druck auf „Cursor nach unten" gelangt man beispielsweise vom Anfang eines Loops an sein Ende, auch wenn dazwischen mehrere Bildschirmseiten Text liegen. Durch Drükken eines Hotkey wird der Compiler aufgerufen und gestartet, der den Cursor auf fehlerhafte Quelltextstellen aufsetzt und die Art des Fehlers angibt. Nach Verlassen des Editors kann man vom Librarian her den Linker starten. Nach den letzten Updates des ITC-Systems war der Editor leider so verschlimmbessert, daß er beim Kopieren von Kommentarzeilen und beim Editieren mehrzeiliger Parameterlisten reproduzierbar kalt abstürzte.

JPI

In eine Umgebung mit Turbo-Pascal-ähnlichen Menüs und einem sehr nützlichen kontext-sensitiven Hilfesystem sind Editor (bis zu vier Fenster gleichzeitig), Maker, Compiler und Linker nahtlos integriert. Compiler und Linker können alternativ auch im Batchbetrieb verwendet werden. Der gesamten Umgebung merkt man an, daß sie von Programmierern für Programmierer gemacht wurde. Fenster, Menüs und Tastenbelegung sind frei gestaltbar. Alle notwendigen Pfade und Suchstrategien für Dateien können über ein sogenanntes Redirection-File sehr komfortabel festgelegt werden. Damit entfällt die Notwendigkeit,

sich das DOS-Environment mit einer Vielzahl von Variablen zu belegen. Das Redirection-File ermöglicht bei geschickter Ausnutzung, daß das Entwicklungssystem als einziges der getesteten voll netzwerkfähig wird. Man kann nicht nur von mehreren Arbeitsplätzen aus gleichzeitig mit denselben Programmen arbeiten, sondern sich auch an einer Workstation beliebige Parameter umdefinieren, ohne einen anderen Benutzer zu beeinflussen. Dies ist um so erfreulicher, als JPI selbst über diese verborgenen Qualitäten des Systems nicht informiert war und auch nicht damit wirbt. Nähere Informationen speziell zum Netzwerkbetrieb sind bei den Autoren erhältlich. Das Environment merkt sich von Sitzung zu Sitzung sogar die editierten Module und die letzte Cursorposition in jedem Editorfenster. Man kann also immer direkt an dem Punkt weiterarbeiten, an dem man unterbrochen wurde. Sehr angenehm bei nicht zu großen Programmen ist die Möglichkeit, einfach die Run-Taste zu drücken. Dies bewirkt, daß automatisch ein Maker- und Linkerlauf ausgeführt und das fertige Programm sofort gestartet wird. Sowohl bei Compilerfehlern als auch bei Laufzeitfehlern wird der Editor aufgerufen und der Cursor direkt auf die fehlerhafte Zeile positioniert. Dieses Environment ist eines der wenigen, das selbst verwöhnten Programmierern noch ab und zu ein bewunderndes „Ah" oder „Oh" entlockt. Die Liebe zum Detail zeigt sich neben vielen anderen Stellen auch beim Compiler, der den Fortgang seiner Arbeit für jedes übersetzte Modul mit einer Balkengrafik dokumentiert oder beim Editor, der vor dem Starten eines neu gelinkten Programms automatisch alle noch geladenen Quelltexte auf Festplatte sichert, damit sie bei einem Rechnerabsturz nicht verloren sind.

Logitech und Stony Brook ...

... sind herkömmliche Entwicklungssysteme mit getrenntem Editor, Maker, Compiler und Linker. Darüber kann auch der Versuch nicht hinwegtäuschen, den Compiler über einen EXEC-Aufruf in den Editor zu integrieren. Sie sind dadurch auch für den Anwender interessant, der sich bereits auf einen bestimmten Editor, Linker oder Maker festgelegt hat und sich nicht extra umstellen will.

Logitech

Im Logitech-Entwicklungssystem wird die Benutzeroberfläche über das Erweiterungsprogramm M2ASSIST des POINT-Editors definiert. Mit dem POINT-Editor können in mehreren Bildschirmfenstern Quellcode-Dateien geschrieben werden. Der stark von der Logitech-Maus abhängige Editor spielt bei den ansonsten rein tastenorientierten Programmeditoren eher eine Außenseiterrolle. In dem Kreis der uns bekannten Softwareentwickler wird er nicht in größerem Maßstab eingesetzt, aber besonders in den USA scheint er unter Profis recht beliebt zu sein. Außerdem besteht für Mausfeinde ja immer die Möglichkeit, einen beliebigen anderen ASCII-Editor einzusetzen. Über Funktionstasten kann man Compiler, Linker oder DOS-Shell aufrufen, ohne den POINT-Editor zu verlassen. Mit einer Konfigurationsdatei kann die Leistungsfähigkeit des Systems, die Art der Hilfen und die Tastenbelegung in weiten Grenzen selbst gestaltet werden. Logitech legt alle erzeugten Dateien (Endungen def, mod, sym, ref, map, exe) im aktuellen Subdirectory ab. Die Angabe von besonderen Pfaden für die Ausgabedateien von Editor, Compiler und Linker sind nicht vorgesehen. Bei einem größeren Projekt können so schon mal 400 Dateien in einem Subdirectory bunt gemischt stehen.

Stony Brook

Hier existiert kein geschlossenes Environment. Der mitgelieferte Editor ist zwar sehr kompakt, besitzt aber genauso wie alle anderen getesteten Editoren die wichtigsten Fähigkeiten, die bei der Programmerstellung benötigt werden: Compileraufruf aus dem Editor mit Fehleranzeige im Sourcecode, split-screen-editing, verschiedene Fenster und Aufruf der DOS-Shell. Außerdem wird er im (Modula-)Sourcecode geliefert und kann somit gut den eigenen Gewohnheiten angepaßt werden. Der Edit-Compile-Link-Go-Zyklus läuft völlig unspektakulär ab und unterscheidet sich nicht von anderen Batch-orientierten Compilern. Selbstverständlich erkennt auch hier, wie bei allen getesteten Entwicklungssystemen, der Maker die Modulabhängigkeiten anhand der Importlisten der einzelnen Module selbst. Man braucht also nicht (wie bei den gängigen C-Make-Utilities) diese Abhängigkeiten in einer Textdatei selbst aufzulisten. Da kein eigener Linker mitgeliefert wird, muß man auf den DOS-Linker oder einen anderen kompatiblen Linker zurückgreifen. Dabei sollte man unbedingt darauf achten, daß der Linker seine Eingaben wie der DOS-Linker aus einem sogenannten Linker-Response-File beziehen kann, da sonst die Make-Utility nicht funktioniert. Der Nachteil eines externen Linkers besteht darin, daß er nur modulweise linken kann, während gute Modula-2-Linker (Logitech und JPI) prozedurorientiert arbeiten und auf diese Weise nicht ein ganzes Modul dazulinken, nur weil eine Prozedur benötigt wird.

Compiler und Compilerfeatures

Die im folgenden gegebenen Geschwindigkeitswerte wurden auf einem Compaq 386/20 ohne RAM-Disk oder Festplatten-Cache ermittelt. Während der täglichen Arbeit in unserem Ethernet-Netzwerk ergeben sich bessere Zeiten, da hier ein Dedicated Server mit mehreren MByte RAM zur Verfügung steht und alleine das Novell-Filesystem schon um den Faktor 3 schneller ist als das MS-DOS 3.3 Filesystem. Alle hier getesteten Compiler und Linker würden in ähnlicher Weise von einem Cache oder einer großen RAM-Disk profitieren, aber für Testzwecke ist die Beschränkung auf den nackten Rechner ohne Netzwerk, RAM-Disk oder Cache gerechtfertigt.

ITC

Der Compiler wird vom Editor unterstützt: Weil der Editor intern keinen ASCII-Text speichert, sondern schon während der Eingabe des Programms den Parse-Tree aufbaut, entfallen für den Compiler Syntaxprüfung und Parsing. Für unseren Benchmark, die Übersetzung der Utility 'GO.MOD' mit 400 Zeilen Sourcetext, braucht der Compiler ca. 6,8 Sekunden. Wie bei allen andern Compilern hängt die Geschwindigkeit im Einzelfall stark davon ab, wieviele andere Module das übersetzte Modul importiert. Der Grund dafür ist die komplette Typprüfung und Versionskontrolle auch über Modulgrenzen hinweg, die jeder Modula-2-Compiler während des Übersetzungsvorgangs durchführen muß, um eventuelle Inkonsistenzen zu erkennen. Der Compiler ist voll ins Environment integriert und kann nicht im Batch-Betrieb verwendet werden. Er verfügt im Vergleich zu den anderen Compilern über die wenigsten Optionen und überhaupt keine Direktiven. Die einstellbaren Optionen betreffen Stack-, Range- und Arithmetik-Checks. Die erzeugten linkfähigen Segmente haben ein eigenes Format und können nicht mit einem fremden Linker verarbeitet werden. Dadurch gibt es auch keine Möglichkeit, Code von anderen Compilern in das Modula-Programm mit einzubinden. Aufgrund seines Alters spiegelt der ITC-Compiler im wesentlichen noch die Sprachdefinition der 1. Ausgabe von Wirths „Programmieren in Modula-2" wieder. Manche Möglichkeiten, die neuere Compiler bieten, fehlen hier noch. Dazu zählt beispielsweise die Mög-

lichkeit, eigene Terminate-Handler und Error-Handler zu installieren. Bei der Formulierung von Interruptprozessen gibt es eine kleine Abweichung von der Standardsyntax. Unterschiedliche Modulprioritäten werden nicht unterstützt. ITC arbeitet stets mit dem Code-Modell „large". Zusätzlich zu den sonst in Modula-2 üblichen ARRAY OF CHAR unterstützt ITC als einziger der getesteten Compiler auch Strings im Turbo-Pascal-Format (Länge in Byte 0). Inline-Assemblercode kann mit CODE-Statements angeben werden. Externe Prozeduren können mit einigen Einschränkungen verwendet werden. Dazu dient ein sogenannter FOMI (Foreign Object Module Importer), der aber den Import nur einer Prozedur je Modul erlaubt. Wer auf Portabilität achten muß, sollte Features, die vom Standard abweichen, generell mit Vorsicht einsetzen.

JPI

Die Compiliergeschwindigkeit kann sich trotz Optimierung sehen lassen: 15 000 Zeilen pro Minute auf dem hier verwendeten Compaq 386/20 sind für Modula-2 ein mehr als ausreichender Wert, zumal nach einer Programmänderung meist nur einzelne Module neu übersetzt werden müssen. Unser Testprogramm übersetzt JPI in 3,4 Sekunden. In der Praxis sind Maker, Compiler und Linker so schnell, daß man selbst bei einem Projekt mit 200 KByte *.EXE File zwischendurch öfters mal einen Testlauf startet, indem man einfach die 'RUN' Taste drückt und alles andere dem Entwicklungssystem überläßt. Ein Teil der Compiliergeschwindigkeit wird dadurch erkauft, daß es sich um einen Single-Pass-Compiler handelt. Die damit einhergehenden Nachteile (u. a. die Notwendigkeit von FORWARD-Deklarationen) sind in der Beurteilung am Ende dieses Artikels zusammengefaßt. Allerdings befindet sich JPI hier in guter Gesellschaft: Auch Wirths aktuellste Compilerversion ist ein Single-Pass Compiler. Die erzeugten *.OBJ Dateien können bei Bedarf mit einem anderen Linker als dem eingebauten verarbeitet werden. Die Handhabung ist dann aber wieder genauso umständlich, wie vom DOS-Linker gewohnt, und das erzeugte *.EXE-File ist im allgemeinen größer, da nur der JPI-Linker automatisch die nicht benötigten Prozeduren jedes Moduls beim Linken entfernt. Dazu mehr im Abschnitt über die verschiedenen Utilities.

Der Sprachumfang des Compilers hält sich streng an die aktuellste Definition von Wirth. Darüber hinaus gibt es die zusätzlichen Datentypen BYTE, LONGWORD, SHORTCARD, LONGCARD, SHORTINT und LONGREAL.

Mit verschiedenen Compilerdirektiven kann man die Codeerzeugung so steuern, daß das Linken mit *.OBJ-Dateien aus anderen Sprachen und Speichermodellen ermöglicht wird. Für den Aufruf von C-Routinen veranlaßt die Direktive (*$K*), daß die Parameter wie bei C üblich in umgekehrter Reihenfolge auf den Stack gelegt und vom aufrufenden Programm wieder entfernt werden. Der Compiler sieht keine Möglichkeit vor, eigenen Inline Code zu verwenden. Wichtige Funktionen wie Enable/Disable Interrupt stellt das Modul SYSTEM als Prozeduren zur Verfügung. Der Compiler setzt diese Prozeduren dann nicht in Aufrufe ans Runtime-System, sondern in schnellen Inline-Code um.

Logitech

Der Logitech-Compiler erledigt seine Aufgabe gründlich, aber nicht übermäßig schnell, obwohl gegenüber der Version 2.0 gravierende Verbesserungen vorgenommen worden sind. Er bietet dafür als einziger der hier verglichenen Compiler zwei verschiedene Compilerversionen: Eine voll gelinkte und eine Overlay-Version. Mit der Overlay-Version kann man größere Programme übersetzen, da mehr dynamischer Speicherplatz für die Symboltabellen frei bleibt, oder man kann trotz einiger geladener TSR-Programme noch Compilerläufe durchführen. Unser Benchmark ergab 8,2 Sekunden für die Overlay- und 6,0 Sekunden für die voll gelinkte Version. Bei abgeschalteter Optimierung erhöht sich die Geschwindigkeit etwas.

Der Compiler hält sich weitgehend an den Standard. Darüber hinaus wird Adreßarithmetik in begrenztem Umfang unterstützt. Der Typ PROCESS ist noch extra definiert, obwohl Wirth inzwischen auf die Unterscheidung von PROCESS und ADDRESS verzichtet hat. Es werden nun im Gegensatz zur Version 2.0 Standard *.OBJ-Dateien erzeugt, so daß auch hier das Interfacing zu anderen Sprachen möglich ist. In diesem Bereich geht die gebotene Unterstützung weit über das von anderen Compilern gewohnte Maß hinaus. Ein Programm kann beispielsweise aus einem Hauptprogramm in Microsoft C und Unterprogrammen aus Logitech Modula bestehen, oder umgekehrt. Die gleichzeitige Verwendung beider Runtime-Systeme ist möglich, sofern das jeweils fremde Runtime-System explizit initialisiert wird. Das Logitech-Handbuch gibt zu diesem Thema viele Hinweise und wertvolle Tips und unterstreicht damit den professionellen Anspruch des Entwicklungssystems.

Stony Brook

Der Compiler brauchte für unser Testprogramm nicht einmal 2 Sekunden zur Compilation. Er erreichte eine Spitzengeschwindigkeit von 20 000 Zeilen pro Minute. Wenn auch die reine Compilergeschwindigkeit bei der Entwicklung einer Applikation nicht sehr stark ins Gewicht fällt, trägt dies doch zum guten Gesamteindruck bei, vor allem da der erzeugte Code bereits gut optimiert ist.

Auch hier werden, wie bei JPI und Logitech, Standard *.OBJ Dateien erzeugt. Der Sprachumfang nach Wirth wird vom Compiler voll unterstützt. Inline Code kann nicht verwendet werden. Für die Verbindung von Modula-2 Programmen mit *.OBJ-Modulen aus anderen Sprachen gibt es die Möglichkeit, externe Prozeduren durch die Angabe von Attributen genau zu charakterisieren. So kann man beispielsweise deklarieren, welche Register von einer externen Prozedur verändert werden, ob eine feste oder variable Anzahl von Parametern erwartet wird, wie die Parameterübergabe stattfindet, in welchen Registern Funktionsergebnisse zurückgegeben werden und vieles mehr. Damit sollte es kein Problem sein, selbst umfangreiche, in C geschriebene Bibliotheken in ein Modula-2-Programm einzubinden. Stony Brook hat diese Notwendigkeit klar erkannt und schreibt in der Dokumentation über diese Features u. a.:

„Der Stony Brook Modula-2 Compiler wurde mit dem Gedanken entwickelt, Programmiersprachen zu kombinieren. Es gibt eine Menge Softwarepakete für MS-DOS-Rechner, und bedauerlicherweise sind die meisten davon nicht in Modula-2 geschrieben. Mit dem Stony Brook Modula-2-Compiler kann man Programmbibliotheken verwenden, die in anderen Sprachen geschrieben sind." (Übersetzt aus dem Stony-Brook-Handbuch, S. 87).

Stony Brook kennt sechs verschiedene Speichermodelle, die auch gemischt werden können. Der Compilervorgang kann mit insgesamt 18 verschiedenen „Qualifiers" (Compilerswitches) gesteuert werden. Die interessantesten betreffen: 6 verschiedene Runtime-Checks, 2 Optimierungsmöglichkeiten sowie die Festlegung der zu verwendenden Segmentnamen. Die Größe der übersetzbaren Programme hängt nur vom verfügbaren Speicherplatz zur Compilierzeit ab. Stony Brook gibt an, daß der Compiler nur ca. 90 KByte dynamischen Speicherplatz braucht, um sich selbst zu übersetzen und zeigt, im Gegensatz zu den anderen Anbietern, wie Speicherplatz bei der Compilierung zu sparen ist.

Codequalität und Compilerfehler

Die Qualität des erzeugten Codes ist bei den einzelnen Compilern sehr unterschiedlich. *Bild 1 und 2* zeigen zwei kleine Testprogramme, die wir mit allen Compilern übersetzt haben. Wo möglich, haben wir die Compiler-Option „Optimierung" verwendet. Den erzeugten Code haben wir vom Runtime-System befreit und kommentiert (Listing 1a-1d und 2a-2d). Das erste Programm zeigt, wie effektiv Schleifen und Prozeduraufrufe übersetzt werden, das zweite ist ein Beispiel für Longinteger-Arithmetik, bei dem die Compiler von ITC und Logitech bereits passen mußten. Für beide Tests haben wir alle Compiler-Checks abgeschaltet, damit der erzeugte Code übersichtlich bleibt. Auf die unterschiedliche Qualität der Checks bei den einzelnen Compilern kommen wir ein paar Abschnitte weiter noch zu sprechen.

ITC

Weder für Schleifenvariable noch für lokale Variable werden Register verwendet. Es findet keine Optimierung statt. Der Compiler verfügt nicht über die Möglichkeit des 'Register-Bookkeeping', lädt also häufig Register mit einem Wert nach, den sie bereits enthalten (Beispiel: MOV DS,CS:[0000] an mehreren Stellen in Listing 1a). Auffällig ist der wenig elegante Code für die einfache FOR-Schleife mit einem Vergleich am Anfang und einem zweiten am Ende der Schleife. Für das Programm aus Listing 2 konnte kein Code erzeugt werden, da der Compiler den Ausdruck

```
a := b*c – d*e
```

mit der Fehlermeldung „Expression too complicated" ablehnt. Selbstverständlich könnte man den Ausdruck berechnen, indem man ihn in zwei Teile aufspaltet; dies ist aber nicht der Sinn des Vergleichs. Der Compiler erzeugt aber immerhin für die Multiplikation mit 2 nur einen Schiebebefehl. Er unterscheidet in der Codeerzeugung noch nicht zwischen 8088 und 80286 als Zielprozessor, was angesichts seines Alters nicht verwundern darf.

JPI

Dieser Compiler setzt den Maßstab für die Codequalität. Er verwendet stets so viele Register wie möglich, führt über deren Belegung Buch und sichert vor einem Prozeduraufruf nur die Register, die auch wirklich in der Prozedur verändert werden. Für Schleifen erzeugt er einen Code, wie man ihn als Assemblerprogrammierer erwarten würde. Wir haben in einem anderen Test eine 9fach geschachtelte Schleife übersetzen lassen: Der Compiler verwendete zuerst alle acht verfügbaren Register und erst dann eine Speichervariable für die neunte Schleife. Die Longint-Arithmetik ist effizient und fehlerfrei. Dabei ist interessant, daß der Compiler auch für Teile einer Variablen (hier für das höherwertige Wort von Longinteger-Variablen) ein Register verwendet, selbst wenn für die ganze Variable nicht genügend Register frei sind. Im erzeugten Code wird nur der 8086-Befehlssatz verwendet, eine Option zur Umschaltung auf 80286 ist nicht vorhanden. Ein JPI-Mitarbeiter, mit dem wir uns über das Thema Codequalität unterhalten haben, meinte trocken:
„Da steckt auch eine ganze Menge an Optimierung drin."
Das Ergebnis gibt ihm recht.

Logitech

Die beiden Optimierungsmöglichkeiten des Compilers bestehen im 'Register Bookkeeping' und in der Verwendung von 80286 Instruktionen. In unseren Listings fallen vor allem die Enter- und Leave-Befehle auf, mit denen der 80286 gezielt Hochsprachencompiler unterstützt. Ansonsten ist der erzeugte Code eher durchschnittlich, da zu wenig mit den Prozessorregistern gearbeitet wird. Die Multiplikation mit 2 wird korrekt in einen Schiebebefehl umgesetzt. Die in der Version 3.0 dazugekommene Longinteger-Arithmetik scheint den Compiler zu überfordern. Der erzeugte Code für unser zweites Testprogramm ist umständlich und fehlerhaft: Nach der Multiplikation c*d liegt das 32-Bit-Ergebnis in den Registern AX und BX vor. Danach wird die zweite Multiplikation durchgeführt, ohne AX und BX zu sichern. Zum Schluß subtrahiert der erzeugte Code dann AX von AX und BX von BX. Das Ergebnis ist natürlich immer 0 ! In einer unserer Datenbankanwendungen führte dieser Fehler dazu, daß die Signatur einer Datei nicht korrekt berechnet wurde und sich eine Datei nach dem Anlegen nie wieder öffnen oder bearbeiten ließ... Dieser Fehler im Codegenerator wird voraussichtlich in der nächsten Compilerversion beseitigt.

Stony Brook

Der Compiler erzeugt, soweit wir bisher feststellen konnten, fehlerfreien Code. Er setzt die Prozessorregister sinnvoll ein und lädt sie nicht unnötig oft mit bereits vorhandenen Werten nach. Im ersten Test (Listing 1d) erzeugt er einen geringfügig besseren Code als der JPI Compiler: Er erkennt, daß in der Prozedur Use() der Basepointer BP nicht verändert wird und spart den Befehl MOV SP,BP vor dem Rücksprung ein. Die Multiplikation setzt auch er in einen einfachen Schiebebefehl um. In unserem zweiten Beispiel erzeugt Stony Brook einen geringfügig längeren Code als JPI, weil er die Register nicht so intensiv einsetzt, so daß der Vergleich doch wieder unentschieden endet. Um die Ergebnisse hier besser beurteilen zu können, muß man die Geschwindigkeit des erzeugten Codes im Vergleich mit dem eines guten C-Compilers betrachten. Stony Brook schneidet hier in den verschiedenen bekannten Benchmarks durchschnittlich 20–30 % besser ab, als Borlands Turbo-C.

Fließkomma-Arithmetik

Alle Compiler arbeiten mit Real-Variablen im 64-Bit-IEEE-Format. JPI und Stony Brook nennen solche Zahlen LONGREAL und bieten zusätzlich 32-Bit-Reals unter dem Typbezeichner REAL an. Hier wäre es aus Gründen der Kompatibilität günstiger gewesen, REAL mit 64 Bit zu implementieren und den 32-Bit-Typ mit SHORTREAL zu bezeichnen. Logitech bietet darüber hinaus noch BCD-Arithmetik mit beliebig großer Stellenzahl an. Alle Compiler unterstützen einen Numerik-Coprozessor. Bei ITC wird zur Laufzeit auf den Coprozessor oder den Emulator verzweigt. JPI erzeugt Code für den Coprozessor und patcht diesen Code zur Laufzeit einmal um, wenn nur der Emulator vorhanden ist. Alternativ kann auch bereits beim Linken Code erzeugt werden, der nur mit Coprozessor lauffähig ist. Logitech liefert verschiedene Real-Bibliotheken mit, so daß man wahlweise Programme erzeugen kann, die nur mit Coprozessor lauffähig sind oder solche, die automatisch zwischen Emulator und Coprozessor umschalten. Der Vorteil bei Programmen, die nur mit Coprozessor lauffähig sein müssen, besteht darin, daß der Compiler in diesem Fall die Coprozessorbefehle als Inline Code einfügen kann, was einen weiteren Geschwindigkeitsvorteil mit sich bringt. Stony Brook verwendet einen Compilerswitch, mit dem die Codeerzeugung für den Coprozessor gesteuert wird. Ein Programm, das für Coprozessor compiliert wurde, kann nicht auf einem Computer ohne Coprozessor ablaufen.

Maker

Bei ITC ist der Maker in die Umgebung integriert. Er ermittelt die Modulabhängig-

Listing 1: Test der Codeerzeugung

```
MODULE CodeGen1;

VAR
  i: CARDINAL;

PROCEDURE Use (n: CARDINAL);
  VAR
    n2: CARDINAL;
  BEGIN
    n2 := 2*n
  END Use;

BEGIN
  FOR i:=1 TO 30000 DO
    Use (i)
  END
END CodeGen1.
```

Listing 1a: Erzeugter Code (ITC):

```
;
; PROCEDURE Use(n: CARDINAL);
;
2AD9:0A18  PUSH  BP              ;Zugriff auf Parameter
2AD9:0A19  MOV   BP,SP
2AD9:0A1B  PUSH  AX              ;Ausgleich für RET Far
2AD9:0A1C  MOV   AX,[BP+04]      ;Parameter holen
2AD9:0A1F  SHL   AX,1            ;Multiplikation mit 2
2AD9:0A21  MOV   [BP-02],AX      ;Ergebnis in lokale Variable
2AD9:0A24  MOV   SP,BP           ;typisches Prozedur-Ende mit
2AD9:0A26  POP   BP              ;Parameterbeseitigung
2AD9:0A27  RET   Far 0002        ;vom Stack
;
; Hauptprogramm
;
2AD9:0A2A  PUSH  BP              ;Hauptprogramm wird wie eine
2AD9:0A2B  MOV   BP,SP           ;Prozedur behandelt !
2AD9:0A2D  MOV   DS,CS:[0000]    ;Datensegment einrichten
2AD9:0A32  MOV   [0000],0001     ;Schleife mit 1 beginnen
  Loop:
2AD9:0A38  MOV   DS,CS:[0000]    ;DS neu setzen (ist sicherer)
2AD9:0A3D  CMP   [0000],7530     ;auf Schleifenendwert testen
2AD9:0A43  JA    Done
2AD9:0A45  PUSH  [0000]          ;Parameterübergabe an Use()
2AD9:0A49  CALL  0A18            ;Use(i)
2AD9:0A4C  MOV   DS,CS:[0000]    ;(schon wieder...)
2AD9:0A51  ADD   [0000],0001     ;besser wäre INC !
2AD9:0A56  JNC   Loop            ;(wegen 65535 Sonderfall)
  Done:
2AD9:0A58  MOV   SP,BP           ;typische Ende-Sequenz
2AD9:0A5A  POP   BP              ;jeder Prozedur
2AD9:0A5B  RET   Far             ;Rückkehr zum Runtime-System
```

Listing 1b: Erzeugter Code (JPI):

```
;
; PROCEDURE Use(n: CARDINAL);
;
3535:0000  PUSH   BP
3535:0001  MOV    BP,SP          ;Parameter vom Stack holen
3535:0003  MOV    AX,[BP+06]
3535:0006  SHL    AX,1           ;Multiplikation mit 2
3535:0008  MOV    SP,BP          ;unnötig, da SP nicht verändert
3535:000A  POP    BP             ;Parameter vom Stack nehmen
3535:000B  RET    Far  0002      ;und zurückkehren.
;
; Hauptprogramm
;
3535:000E  MOV    DS,CS:[0023]   ;(unnötig)
3535:0013  MOV    CX,0001        ;FOR-Schleife starten
3535:0016  PUSH   CX             ;Parameter auf den Stack
3535:0017  PUSH   CS             ;weil Use() mit RET Far endet
3535:0018  CALL   0000           ;Use(i)
3535:001B  INC    CX             ;Schleife schließen
3535:001C  CMP    CX,7530        ;bis i=30000
3535:0020  JNA    0016
3535:0022  RET    Far            ;Rückkehr zum Runtime System
```

Listing 1c: Erzeugter Code (Logitech):

```
DATA       Segment: CODEGEN1_DATA
           Length : 2
0000  i__CodeGen1   DUP    2

CODE       Segment: CODEGEN1_TEXT
           Length : 91
0000       DB     '..'
;
; PROCEDURE Use(n: CARDINAL);
;
0002       ENTER 0002,01          ;186/286
0006       MOV   AX,[BP+0004]     ;Parameter vom Stack holen
0009       SHL   AX,1             ;Multiplikation mit 2
000B       MOV   [BP-0004],AX     ;Produkt abspeichern
000E       LEAVE                  ;186/286
000F       RET   0002             ;Parameter vom Stack und zurück
;
; Hauptprogramm
;
0012       ENTER 0000,01          ;186/286
0016       MOV   CX,SEG CODEGEN1_DATA  ;Datensegment setzen
0019       MOV   DS,CX
001B       MOV   BX,i__CodeGen1   ;BX für Zugriff auf i
001E       MOV   [BX],0001        ;i mit 1 belegen
0022       DEC   [BX]             ;erstes INC ausgleichen
0024       INC   [BX]             ;Schleifenanfang
0026       PUSH  BX               ;Register vor Use() retten
0027       PUSH  DS
0028       MOV   CX,SEG CODEGEN1_DATA  ;Parameter auf den Stack
002B       MOV   ES,CX
002D       MOV   AX,ES:[i__CodeGen1]
0031       PUSH  AX
0032       CALLS S__Use__CodeGen1 ;Use(i)
0035       POP   DS               ;Register wieder herstellen
0036       POP   BX               ;(dumm)
0037       CMP   [BX], 7530       ;Schleife schließen
003B       JB    0024
003D       LEAVE                  ;186/286
003E       RET                    ;Rückkehr zum Runtime System
```

Listing 1d: Erzeugter Code (Stony Brook):

```
;
; PROCEDURE Use(n: CARDINAL);
;
0000 P0002
0000       PUSH  BP
0001       MOV   BP, SP           ;Parameter vom Stack hölen
0003       MOV   DX, 0006[BP]
0006       SHL   DX, 0001         ;mit 2 multiplizieren
0008       POP   BP
0009       RET   Far 0002         ;Stack aufräumen und zurück
;
; Hauptprogramm:
;
000C       CALL  SYSTEM_Start     ;Initialisierung
0011       MOV   CX, 0001         ;Schleife beginnen
0014 L0003
0014       PUSH  CX               ;Parameter an Use() übergeben
0015       PUSH  CS               ;weil Use() mit RET Far endet
0016       CALL  P0002            ;Use(i)
0019       INC   CX               ;Schleife schließen
001A       CMP   CX, 7530
001E       JBE   L0003
0020       XOR   AX, AX           ;Programm übers Runtime..
0022       PUSH  AX               ;..System beenden
0023       CALL  SYSTEM_HALT
```

Bild 1. Zum Test der Codeerzeugung eine Schleife mit Inhalt

Listing 2: Test der Codeerzeugung

```
MODULE CodeGen2;

VAR
  a,b,c,d,e: LONGINT;

BEGIN
  b := 100000;
  c := VAL (LONGINT,256);
  d := 65536;
  e := VAL (LONGINT,3);
  a := b*c-d*e (* Soll: 25.403.392 *)
END CodeGen2.
```

Listing 2a: Erzeugter Code (ITC):

Kein Code erzeugt wegen "Expression too complicated". Siehe Text.

Listing 2b: Erzeugter Code (JPI):

```
1E8E:0450  CS:                      ;Datensegment definieren
1E8E:0451  MOV   DS,[0041]
1E8E:0455  MOV   SI,86A0            ;b := 100000
1E8E:0458  MOV   AX,0001            ;     (Reg. SI,AX)
1E8E:045B  MOV   DI,0100            ;c := VAL (LONGINT,256)
1E8E:045E  SUB   DX,DX              ;     (Register DI,DX)
1E8E:0460  AND   WORD PTR [000C],+00   ;d := 65536
1E8E:0465  MOV   CX,0001            ;     ([000C],CX)
1E8E:0468  MOV   WORD PTR [0010],0003  ;e := VAL (LONGINT,3)
1E8E:046E  SUB   BX,BX              ;     ([0010],BX)
1E8E:0470  PUSH  AX                 ;b auf den Stack für Mult.
1E8E:0471  PUSH  SI
1E8E:0472  PUSH  DX                 ;c auf den Stack für Mult.
1E8E:0473  PUSH  DI
1E8E:0474  CALL  1E8E:0395          ;Multiplikation b*c mit
1E8E:0479  MOV   DI,AX              ;Ergebnis von (AX,DX)
1E8E:047B  MOV   SI,DX              ; retten nach (DI,SI)
1E8E:047D  PUSH  CX                 ;d auf den Stack für Mult.
1E8E:047E  PUSH  [000C]
1E8E:0482  PUSH  BX                 ;e auf den Stack für Mult.
1E8E:0483  PUSH  [0010]
1E8E:0487  CALL  1E8E:0395          ;Multiplikation d*e
1E8E:048C  SUB   DI,AX              ;b*c - d*e nach (DI,SI)
1E8E:048E  SBB   SI,DX
1E8E:0490  RET   Far                ;Rückkehr ins RTS
```

Listing 2c: Erzeugter Code (Logitech):

(Siehe Text - der erzeugte Code ist unsinnig und das Rechenergebnis bei der Subtraktion zweier LONGINT-Ausdrücke immer gleich 0 !)

```
DATA        Segment: CODEGEN2_DATA
            Length : 20

0000  a__CodeGen2                DUP       4
0004  b__CodeGen2                DUP       4
0008  c__CodeGen2                DUP       4
000C  d__CodeGen2                DUP       4
0010  e__CodeGen2                DUP       4


CODE        Segment: CODEGEN2_TEXT
            Length : 164

0000        DB    '..'
0002        ENTER 0000,01              ; 186/286
0006        MOV   CX,SEG CODEGEN2_DATA ;Datensegment definieren
0009        MOV   DS,CX
000B        MOV   [b__CodeGen2],86A0   ;b := 100000;
0011        MOV   [0006],0001
0017        MOV   [c__CodeGen2],0100   ;c := VAL (LONGINT,256);
001D        MOV   [000A],0000
0023        MOV   [d__CodeGen2],0000   ;d := 65536
0029        MOV   [000E],0001
002F        MOV   [e__CodeGen2],0003   ;e := VAL (LONGINT,3);
0035        MOV   [0012],0000
003B        MOV   DX,[b__CodeGen2]     ;b auf Stack für Mult.
003F        MOV   AX,[0006]
0042        PUSH  AX
0043        PUSH  DX
0044        MOV   SI,[c__CodeGen2]     ;c auf Stack für Mult.
0048        MOV   BX,[000A]
004C        PUSH  BX
004D        PUSH  SI
004E        CALLF LMult__Longs         ;b*c berechnen
0053        MOV   CX,SEG CODEGEN2_DATA ;Datensegment neu def.
0056        MOV   DS,CX
0058        MOV   SI,[d__CodeGen2]     ;d auf Stack für Mult.
005C        MOV   DX,[000E]
0060        PUSH  DX
0061        PUSH  SI
0062        MOV   CX,[e__CodeGen2]     ;e auf Stack für Mult.
0066        MOV   DI,[0012]
006A        PUSH  DI
006B        PUSH  CX
006C        CALLF LMult__Longs         ;d*e berechnen
0071        SUB   AX,AX                ;"jeder rechnet anders"
0073        SBB   BX,BX
0075        MOV   CX,SEG CODEGEN2_DATA ;Datensegment...
0078        MOV   DS,CX
007A        MOV   [a__CodeGen2],AX     ;"Ergebnis" nach a
007D        MOV   [0002],BX
0081        LEAVE                      ;wie üblich rausgehn
0082        RET
```

Listing 2d: Erzeugter Code (Stony Brook):

```
0000        CALL    SYSTEM_Start       ;Initialisierung (RTS)
;
; Hier stand die Initialisierung des Datensegments mit FFFF
; wurde entfernt, da nur optional im Code enthalten.
;
0013        MOV   _DATA+0004, 86A0 ;b := 100000
0019        MOV   _DATA+0006, 0001
001F        MOV   _DATA+0008, 0100 ;c := VAL (LONGINT,256)
0025        MOV   _DATA+000A, 0000
002B        MOV   _DATA+000C, 0000 ;d := 65536
0031        MOV   _DATA+000E, 0001
0037        MOV   _DATA+0010, 0003 ;e := VAL (LONGINT,3)
003D        MOV   _DATA+0012, 0000
0043        PUSH  _DATA+000A       ;c als 2. Parameter an MULT.
0047        PUSH  _DATA+0008
004B        MOV   AX, _DATA+0004   ;b als 1. Parameter an MULT.
004E        MOV   DX, _DATA+0006
0052        PUSH  DX
0053        PUSH  AX
0054        CALL  LONG_MUL         ;b*c ausrechnen
0059        MOV   BX, DX           ;Ergebnis in Registern halten
005B        MOV   CX, AX
005D        PUSH  _DATA+0012       ;zweite Multiplikation genauso
0061        PUSH  _DATA+0010
0065        MOV   AX, _DATA+000C
0068        MOV   DX, _DATA+000E
006C        PUSH  DX
006D        PUSH  AX
006E        CALL  LONG_MUL
0073        SUB   CX, AX           ;Subtraktion low
0075        SBB   BX, DX           ;Subtraktion high
0077        MOV   _DATA+0000, CX   ;Ergebnis in Variable a
007B        MOV   _DATA+0002, BX

007F        XOR   AX, AX           ;Programm beenden
0081        PUSH  AX               ;mit Hilfe des RTS
0082        CALL  SYSTEM_HALT
```

Bild 2. Ein zweites Testprogramm mit erzeugtem Code

keiten nicht bei jedem Aufruf neu, sondern liest sie aus einer gespeicherten Abhängigkeitsliste, die nur durch Aufruf von 'Depend' neu erstellt wird. Vergißt dies der Benutzer nach der Änderung einer Importliste, geht der Maker von falschen Voraussetzungen aus und compiliert eventuell zu wenige Module. Dies führt dann beim anschließenden Linken zum Abbruch wegen eines Versionskonflikts.
Beim Aufruf von Depend wird neben der Abhängigkeitsliste in der Bibliothek auch eine Textdatei mit einer Liste und einer Matrix der Modulabhängigkeiten erzeugt. Besonders die Matrix ist eine Hilfe beim Analysieren von großen Programmsystemen und wird in dieser Form von keinem der anderen Entwicklungssysteme geboten.
Da der Maker nur einen Tastaturmakro erzeugt, muß man vor jedem Make peinlich darauf achten, daß im Librarian-Fenster der Menübalken auf die Bibliothek zeigt, in der Make aufgerufen wird. Sonst werden die zu kompilierenden Module in der aktuellen Bibliothek neu angelegt. Dadurch kommt beim nächsten Mal der Maker selbst durcheinander. Weder der Maker noch seine Extras sind im Handbuch ausreichend dokumentiert, was bestimmt schon einigen Programmierern schlaflose Nächte bereitet haben dürfte. Das Problem der Versionskonflikte durch nicht zusammenpassende *.DEF-Dateien existiert bei JPI nicht, da *.DEF-Dateien überhaupt nicht kompiliert, sondern bei Bedarf stets im Quellcode gelesen werden. Dies ist aufgrund des schnellen Compilers möglich und führt in Zusammenarbeit mit dem integrierten Maker zu einer äußerst problemlosen Handhabung: Im Normalfall wird man einfach den Run-Befehl geben und dem Maker den Rest überlassen. Der Maker funktioniert bei Logitech und Stony Brook so, wie man dies von anderen Make-Utilities kennt: Er erzeugt eine Batchdatei, die Compiler und Linker mit den notwendigen Argumenten nacheinander aufruft. Der ganze Vorgang kann durch eine Konfigurationsdatei beeinflußt werden, in der global gültige Compileroptionen stehen. Bei Logitech besteht die Möglichkeit, bestimmte Module explizit von der Neucompilation auszuschließen.

Bibliotheken und Runtime-System

ITC bietet neben den üblichen Bibliotheksmodulen wie ASCII, Files, InOut, MathLib0, NumConv, RealConv, RealInOut, Storage und Terminal einige Erweiterungen im Bereich Strings, Graphik (rudimentär, CGA), MS-DOS-Unterstützung, RS232 und große Sets. Die Qualität der Bibliotheksmodule ist gemischt. Das wichtigste, Storage, ist in der vorliegenden Version praktisch unbrauchbar, da viel zu langsam. Eine korrigierte, schnelle Version steht aber unter anderem von den Autoren zur Verfügung. Das Filesystem ist recht langsam und stürzt bei einer zu großen Anzahl von vorhandenen Laufwerken ab. Die MS-DOS-Unterstützung ist gut, so daß man ohne Probleme ein eigenes, auch netzwerkfähiges Filesystem schreiben kann. Weder ein Time-sliced-Scheduler noch ein Modul zur Verwaltung von parallelen Prozessen ist standardmäßig vorhanden. Man ist also auf Low-Level-Funktionen des Compilers angewiesen, um Multitasking zu realisieren. JPI liefert zwei verschiedene Gruppen von Bibliotheksmodulen mit: Zum einen die Standardmodule gemäß Wirth, zum andern JPI spezifische Module mit wesentlich höherer Funktionalität. Parallele Prozesse werden voll unterstützt und ein time-sliced-Scheduler für echtes Multitasking gehört mit dazu. Die Krönung des Ganzen ist ein sehr komfortabler Window-Manager, wie man ihn sonst nur in extra zu kaufenden Bibliotheken findet. Was den Umfang der mitgelieferten Programmbibliothek angeht, liegt Logitech mit 64 Modulen klar an der Spitze. Besonders interessant für unseren Vergleich sind Module, die bei anderen Entwicklungssystemen nicht zur Verfügung stehen. Dazu zählt neben der bereits erwähnten BCD-Arithmetik ein kompletter Satz von Datums-, Zeit-, und Timer-Funktionen, Bit-Block-Operationen, Zugriff auf die MS-DOS Speicher-Managementfunktionen, vollständige Grafikroutinen (leider nur für CGA), und ein Treiber für die Logitech Maus. Von den restlichen Modulen sei hier nur kurz „Processes“ erwähnt, das man gemäß Wirths Vorschlägen implementiert findet. Bei Stony Brook sind die üblichen Bibliotheksmodule sowie Erweiterungen für die über den Standard hinausgehenden Datentypen vorhanden. Für die Einstellung der verschiedenen Betriebsarten des Coprozessors steht ein eigenes Modul zur Verfügung. Multitasking wird nicht mit speziellen Modulen unterstützt. Das Filesystem verfügt über Stream- und Block-I/O. Ein Demoprogramm zeigt am Beispiel vieler (quasi-)gleichzeitig auf dem Bildschirm fahrender Autos die Programmierung von parallelen Prozessen. Der komfortable Zugriff auf MS-DOS-Verzeichnisse wird durch ein eigenes Modul ermöglicht.
Den Umfang und die Qualität der mitgelieferten Bibliotheken sollte man als Programmierer nicht überbewerten. In der Praxis schreibt man im Lauf der Zeit einen Großteil davon wahrscheinlich neu, um eigene Vorstellungen zu verwirklichen. Viel wichtiger sind die Bibliotheken, um daraus zu lernen und Anregungen zu gewinnen.

Die Dokumentation

Zur Zeit ist das ITC-Entwicklungssystem das einzige mit einem deutschen Handbuch. Es umfaßt 260 Seiten im DIN-A5-Format und ist blau auf kopierunfreundlichem Umweltschutzpapier gedruckt. Ein kleines Tutorial führt in die Bedienung der Benutzeroberfläche und des syntaxgesteuerten Editors ein. Das Handbuch ist leicht lesbar geschrieben, in der Organisation aber durchaus verbesserungswürdig. Ein ausführliches Stichwortverzeichnis erleichtert die Orientierung. Der Handhabung der Make-Utility mit ihren Besonderheiten werden lediglich neun Zeilen gewidmet. Den größten Raum nimmt wie auch bei den Handbüchern der drei anderen Entwicklungssysteme die Beschreibung und das Listing der DEF-Module mitgelieferter Bibliotheken ein. Seit der ersten Auflage Anfang 1986 wurde das Handbuch nicht mehr den Änderungen des Entwicklungssystems angepaßt.
Auf 267 Seiten beschreibt das Owner's Manual von JPI in englischer Sprache Installation, Benutzeroberfläche, Sprachumfang, Compiler und Bibliotheksmodule. Die Sprachbeschreibung wird durch acht Fallstudien ergänzt. Eine Kurzreferenz und ein Index erleichtern die Arbeit mit Compiler und Handbuch. In der Praxis wird man anstelle des Handbuchs das kontext-sensitive Hilfssystem verwenden.
Vier dicke Manuals in englischer Sprache geben bei Logitech erschöpfend Auskunft über den Compiler (User's Manual, 400 Seiten), Point-Editor (160 Seiten), Formatter, Linker, Run-Time-Debugger, Disassembler, Versionsverwaltung, Crossreferencer, Maker und Checker (Toolkit, 180 Seiten) und Turbo Pascal To Logitech Modula-2 Translator (200 Seiten). Der Sprachumfang des Compilers wird nicht beschrieben, dafür findet sich im Anhang des User's Manual eine fünfseitige Bibliographie. Jedes Handbuch hat ein detailliertes Inhaltsverzeichnis und einen weniger detaillierten Index.
Die Dokumentation besteht bei Stony Brook aus drei Teilen mit zusammen 370 Seiten Umfang in einem DIN-A5-Ringbuch. Die drei Teile sind „User's Guide“, „Language Reference“ und „Runtime Library Reference“ betitelt. Jeder Teil ist getrennt numeriert und mit einem eigenen Index versehen. In allen drei Teilen findet man genau die Fakten, die man als Programmierer wirklich benötigt. Auch schwierige Themen werden in angemesse-

ner Tiefe und sehr professionell abgehandelt. Es entsteht nie der Eindruck, daß dem Benutzer irgendeine interessante Information vorenthalten würde. Leider ist wie bei den anderen englischsprachigen Handbüchern keine deutsche Übersetzung angekündigt. Viele Programmierer ziehen aber aus leidvoller Erfahrung mit deutschen Übersetzungen sowieso die englischen Originaltexte vor.

Utilities

ITC liefert mit dem SDS-XP Entwicklungssystem neben dem bereits oben beschriebenen FOMI und dem Ressource Compiler einen sogenannter Debugger mit. Er arbeitet nicht symbolisch und nicht auf Source-Code-Ebene und muß außerdem in das zu testende Programm eingebunden und bei Bedarf von diesem mit einem TRANSFER-Befehl explizit aufgerufen werden. In der Praxis ist er dadurch nur von sehr begrenztem Nutzen, vor allem da auch der eingebaute Disassembler in der uns vorliegenden Version (2.1b, noch nicht freigegeben) abstürzt.

JPI liefert zwei Utility-Programme mit: setheaps zum Einstellen der Heap-Größe in einem *.EXE Programm und objcompr zum Komprimieren von *.OBJ Files. Beide Programme werden nur kurz in einer read.me Datei erwähnt.

Bei Logitech sind die unter der Bezeichnung Toolkit zusammengefaßten Utilities eine Fundgrube für jeden Modula-Programmierer.

Der Sourcecode-Formater lernt anhand eines Beispielprogramms den Stil des Programmierers und kann dann beliebige andere Quelltexte im gleichen Stil formatieren. Dies ist eine unschätzbare Hilfe, wenn mehrere Programmierer an einem Projekt arbeiten und trotzdem ein für alle leserlicher Code entstehen soll. Leider hat der Formatter bei der Druckausgabe noch einen Fehler im Seitenumbruch. Der Linker ist ein optimierender Overlay-Linker, der sich bei großen Programmen und eingeschalteter Optimierung allerdings reichlich Zeit läßt. Dafür ist dann aber im resultierenden *.EXE-File kein toter Code mehr vorhanden, was vor allem bei kleinen Programmen, die aus vielen verschiedenen Modulen jeweils nur wenige Prozeduren importieren, wichtig sein kann. Der Symbolic Run-Time Debugger ist nach unserer Erfahrung ein sehr hilfreiches Werkzeug, versagt aber bei großen Programmen, weil der Speicherplatz zu knapp wird. Man kann zwar mit einer Swap-Option den Speicherbedarf des Debuggers von 275 KByte auf 125 KByte reduzieren, aber das eigentliche Problem bei uns schien mit dem großen MAP-File unserer Applikation zusammenzuhängen. Der Debugger stürzte beim Laden der Symbol-Informationen regelmäßig ab. Möglicherweise hätte man dies mit einer geschickten Kombination der vielen Optionen von Compiler, Linker und Debugger verhindern können, aber die Dokumentation war in diesem Fall nicht ganz ausreichend. Für ganz hartnäckige Fälle, in denen ein Programm sporadisch abstürzt, gibt es noch einen Post-Mortem-Debugger. Er bezieht die Information über das abgestürzte Programm aus einem Post-Mortem-Dump, den das Run-Time-System beim Auftreten eines Programmfehlers schnell noch auf die Festplatte schreibt. So kann man den Programmzustand und die Variableninhalte auch nach einem Absturz noch überprüfen. Bei sehr großen Programmen mit schlimmen Fehlern kommt es allerdings auch vor, daß der Rechner beim Schreiben des Post-Mortem-Dumps kalt abstürzt – nobody is perfect. Der Decoder erzeugt aus einem *.OBJ-File ein komplettes Assemblerlisting mit dem Modula-2 Quelltext als Kommentarzeilen. Damit wurde ursprünglich beispielsweise der weiter oben beschriebene Compilerfehler entdeckt. Es lohnt sich, ab und zu ein Modul auf diese Weise zu disassemblieren, um ein Gefühl dafür zu bekommen, wie der Compiler bei der Codegenerierung arbeitet. Der Versionsverwalter erlaubt es, in einem Quelltext verschiedene Versionen eines Programms gleichzeitig zu codieren und später daraus die jeweils benötigte Version zu gewinnen. Beispielsweise kann man einen Graphiktreiber schreiben, der für verschiedene Karten geeignet ist. Ein großer Teil des Codes ist für alle Karten identisch, so daß es sich nicht lohnt, verschiedene Quelltexte zu haben. Mit dem Versionsverwalter kann man vor dem Compilieren aus der „Überversion" für alle Grafikkarten den jeweils benötigten Quelltext für eine bestimmte Grafikkarte erzeugen. Diese Datei kann sofort nach der Compilierung wieder gelöscht werden. Dies erleichtert die Programmpflege und reduziert den für Quelltexte benötigten Speicherplatz auf der Platte. M2XREF erzeugt Cross-Referenz-Tabellen für Modula Quelltexte oder Listing-Files und kann nützlich sein, um die Verwendung einer bestimmten Variablen oder Prozedur innerhalb eines größeren Programms zu kontrollieren. M2CHECK erfüllt eine ähnliche Funktion wie lint für einen C-Compiler. Er warnt den Programmierer vor vielen Fehlern und Unsauberkeiten, die selbst in einem syntaktisch korrekten Programm noch enthalten sein können. Dies sind beispielsweise Zuweisungen an eine Schleifenvariable, vergessene lokale Deklarationen, deklarierte, aber nicht angesprochene Variable und einiges mehr. Insbesondere kann man damit unnötige Importe finden und beseitigen. Bei dieser Fülle von nützlichen Tools kann es sich lohnen, das Logitech Toolkit selbst dann zu kaufen, wenn man einen anderen Compiler, beispielsweise den von Stony Brook, verwenden möchte. Vorher sollte man allerdings genau klären, welche Utilities mit welchen Compilern zusammenarbeiten. Wir konnten dies bisher nur stichprobenartig tun und würden uns über weitere Erfahrungsberichte freuen.

Zum Entwicklungssystem gehört außerdem ein Turbo-Pascal nach Modula-2 Konverter. Er erfüllt seine Aufgabe ungefähr so gut, wie man es von einem automatischen Übersetzer erwartet: Eine Menge Nacharbeit ist in jedem Fall notwendig. Die mitgelieferten Bibliotheken implementieren fast alle Turbo-Pascal Features in Modula-2 und sind auch anderweitig verwendbar.

Außer dem Editor, Compiler und Maker wird nur noch ein Source-Code-Debugger bei Stony Brook mitgeliefert. Man muß also einen Intel/Microsoft-kompatiblen Linker und nach Möglichkeit einen Library-Manager (z.B. Microsoft lib) verwenden. Mit dem Debugger liegen uns keine Erfahrungen vor, da er noch recht neu ist. Wir werden nach Möglichkeit beim Test der OS/2-Version näher auf ihn eingehen.

OS/2-Unterstützung

Ein Modula-2-Entwicklungssystem, das unter dem Microsoft-Betriebssystem OS/2 im Protected Mode läuft, wird zur Zeit nur von Stony-Brook (zum gleichen Preis wie die MS-DOS-Version) angeboten, eine Standard-Library für die Cross-Entwicklung ist für 50$ (mit Quellcode für 100$) erhältlich.

JPI steht mit seinem Modula-Entwicklungssystem für OS/2 kurz vor der Freigabe. Es arbeitet ebenfalls im Protected Mode.

JPI hat uns telefonisch Hoffnung auf eine Vorabversion dieses Systems gemacht, sodaß wir dann über unsere Erfahrungen mit OS/2 und den beiden OS/2-Entwicklungssystemen in einer späteren Ausgabe noch berichten können. Gespräche mit Vertretern von ITC und Logitech ergaben, daß dort in absehbarer Zeit nur bei Logitech mit OS/2-unterstützenden Compilern gerechnet werden kann.

MS-Windows Unterstützung

Programme, die mit Microsoft Windows arbeiten sollen, müssen bestimmte Konventionen beachten. So muß das Datensegment

Tabelle 1: Die Testdaten

Hersteller	Version	ca. Preis
Interface Technologies	M2SDS XP 2.1b	999,00 DM
Jensen & Partners	JPI Modula-2 1.05	281,00 DM
Logitech	Modula-2 V3.0 Development System	754,00 DM
Stony Brook	Stony Brook M-2 V1.1	884,00 DM

Tabelle 2: Die Adressen der Lieferfirmen

Hersteller	Adresse
Interface Technologies	Interface Technologies Corp. 3336 Richmond, Suite 200 Houston, Texas 77098 Tel. 001 713 523 8422
Jensen & Partners	J&P Ltd. 63 Clerkenwell Rd. London EC1M 5NP Tel. 0044 1 253 4333
Logitech	Logitech USA Corporate Headquarters 6505 Kaiser Drive Fremont, CA 94555 Tel. 001 415 795 8500
Stony Brook	Stony Brook Inc. Forest Road Wilton, NH 03086 Tel. 001 603 654 2525

beim Eintritt und beim Verlassen einer Prozedur festlegbar sein und ein entsprechender Vorspann und Nachspann für jede Prozedur erzeugt werden. JPI verwendet dazu die Compilerdirektiven (*$@ name*) zur Festlegung des Datensegments für folgende Prozeduraufrufe und (*$T+*) für die Generierung des Vor- und Nachspanns. Obwohl man also mit der Kombination dieser zwei Direktiven Windows-Applikationen in Modula-2 schreiben könnte, wird dies von JPI weder empfohlen noch weitergehend unterstützt.

Der Stony-Brook-Compiler bietet für denselben Zweck den Qualifier /WINDOWS, der ebenfalls für die Erzeugung des richtigen Prologs und Epilogs bei Prozeduraufrufen sorgt. Der Qualifier muß wie auch bei JPI die Direktiven für jedes Modul eines Programms angegeben werden, das unter Windows laufen soll.

ITC und Logitech Modula bieten keine Windows-Unterstützung.

Der Support

Der Support für ITC-Modula erfolgt in Europa über A. + L. Meier-Vogt aus der Schweiz. Von dort können Updates normalerweise zum Selbstkostenpreis bezogen werden. Ein Modem für Datenfernübertragung sowie eine UUCP-Net Adresse stehen zur Verfügung. Allerdings ließ sich bei unserer Arbeit mit dem ITC-Entwicklungssystem der direkte Kontakt zu ITC/Houston nicht immer vermeiden. Außerdem sponsort ITC eine Konferenz in BIX, das von Deutschland aus über Datex-P und Tymnet problemlos zu erreichen ist.

Bei JPI ist ein Datenfernübertragungsprogramm ins Environment integriert, mit dem man (1200 Baud Modem vorausgesetzt) automatisch die 24-h-Techline von JPI anwählen kann. Auf verschiedene schriftliche und telefonische Anfragen bei JPI erhielten wir in jedem einzelnen Fall extrem schnell und unbürokratisch eine Lösung für unser Problem. Meistens lag der Antwort noch kostenlos die neueste Compilerversion bei. Wir hatten uns zu diesem Zeitpunkt nicht als Tester zu erkennen gegeben. Jensen & Partners ist noch eine der Firmen, bei denen man die Entwickler ohne Umweg über drei Abteilungen selbst ans Telefon bekommt.

Aufgrund einer Umstrukturierung und personeller Probleme gab es in letzter Zeit etwas Probleme mit der Unterstützung durch Logitech/Schweiz. Dies soll sich spätestens bis zum Sommer mit der Einrichtung des neuen Münchner Büros ändern, wie uns ein Vertreter der Firma im Gespräch mitteilte. Zusätzlich besteht auch hier der direkte Draht zur amerikanischen Niederlassung und einer Logitech Konferenz in BIX. Die entsprechenden Namen, Adressen und Telefonnummern sind auf Anfrage erhältlich. Aus postalischen Gründen können sie leider nicht alle im Rahmen dieser Besprechung abgedruckt werden ...

Stony Brook unterhält eine Hotline und eine BIX-Konferenz. Da der Compiler bisher problemlos lief, konnten wir keine weitergehenden Erfahrungen mit dem Service machen. Wer einen Fehler im Compiler findet, erhält dafür das nächste Update, mit dem der Fehler beseitigt wird, gratis. Das erste Update erhielten wir gratis und unaufgefordert, obwohl wir die Registrierkarte noch nicht zurückgesendet hatten.

Für welches Entwicklungssystem soll man sich entscheiden? Jedes der vier hat ausgeprägte Stärken und Schwächen:

ITC

Dies ist das älteste Entwicklungssystem, mit dem u. a. Programme wie Farsight, Komforttext, KILO und ein Teil der Advanced Library/Net entstanden sind. Es hat die fortschrittlichste Umgebung und einen Editor, der die Sprache wirklich kennt, einen sehr schnellen Compiler, eine ausgezeichnete Bibliotheks- oder Projektverwaltung und ein brauchbares Overlay-Konzept mit Runtime-Linking. Nach wie vor stürzt aber der schöne Editor bei langen Kommentaren oder Parameterlisten kalt ab und erzeugt der Compiler fehlerhaften Code für bestimmte VAL () Statements. Die erzeugten Programme stürzen wegen eines Fehlers im Bibliotheksmodul Files ab, sobald die Workstation-Shell für ein Novell-Netzwerk geladen ist oder die Konfigurationsvariable lastdrive höher als G: gesetzt wird. Die mitgelieferten Bibliotheksmodule entsprechen von der Performance her nicht mehr dem gewohnten Standard. Allerdings sind verbesserte Module z. B. von RTA (Reals) und MTS (alle) erhältlich. Der Maker ist gewöhnungsbedürftig und im Handbuch nur unzureichend erklärt. Da keine Standard *.OBJ-Files erzeugt werden, kann man nicht mit Modulen aus anderen Sprachen linken. Einige Abweichungen vom „Standard", beispielsweise im Umgang mit Interrupts, sollten beseitigt werden. Der sehr schnelle Edit-Compile-Link-Zyklus entschädigt jedoch für einige Unannehmlichkeiten, und nicht zuletzt ist dies das einzige System, zu dem ein deutsches Handbuch mit einem kurzen Tutorial geliefert wird.

JPI

Dieses Entwicklungssystemen ist erst seit einem halben Jahr auf dem Markt. Der Compiler erzeugt Maschinencode im Intel-Format (*.OBJ), der das Prädikat „handcodiert" verdient. Die mitgelieferten Bibliotheksmodule (mit Quellcode) sind ideenreich und effizient programmiert und zeigten sich bisher fehlerfrei. Aus der anpassungsfähigen Benutzeroberfläche ähnlich der von Turbo-Pascal 4.0 werden Editor, Compiler, Linker und Maker bei Bedarf automatisch aufgerufen. Da der Compiler sich sehr eng an Wirths aktuellste Sprachdefinition hält und dieses Entwicklungssystem das billigste der vier getesteten ist, wird man an das Auftauchen des ersten preiswerten Pascal-Compilers vor ein paar Jahren erinnert ... Der schnelle Einpaß-Compiler bedingt leider, daß die Definition nicht exportierter Prozeduren und Variablen vor ihrer ersten Verwendung stehen muß. Dadurch wird dem Programmierer

manchmal eine unübersichtliche Anordnung seines Programmcodes aufgezwungen, falls er keine FORWARD Deklarationen verwenden möchte. Besonders unangenehm fällt dies bei der Übernahme von Quellcode eines anderen Entwicklungssystems auf. Bei einer unserer Anwendungen trat ein Überlauf in der Symboltabelle des Linkers auf: „Sorry: symbol table too large". Die Antwort von JPI mit Tips zur Umgehung des Problems traf wie bereits in einem anderen Fall innerhalb von zwei Tagen aus London ein.

Logitech

Ein sehr solider Compiler (bis auf den Ausrutscher mit der LONGINT-Arithmetik) und die mit Abstand größte Sammlung an Utilities sowie die sehr ausführliche Dokumentation und die große Unterstützung des Systems durch Fremdfirmen lassen Logitech nach wie vor sehr interessant erscheinen. Die Entscheidung für dieses Produkt ist sicher in vielen Firmen leichter zu rechtfertigen, als für Stony Brook oder JPI, die bei uns zu Unrecht noch nicht so bekannt sind. Wer Tools wie den Source-Code-Formatter oder den Cross-Referencer benötigt, kommt an Logitech sowieso nicht vorbei. Allerdings sollte für diese Zielgruppe möglichst schnell die OS/2-Unterstützung angekündigt und verfügbar gemacht werden.

Stony Brook

Der Compiler besticht durch seine Flexibilität bei der Kopplung von Modula-2 mit anderen Sprachen. Er ist schnell, dicht am Standard und erzeugt soweit uns bekannt fehlerfreien, optimierten Code. Die fehlenden Tools schränken die Zielgruppe natürlich etwas ein. Die Dokumentation empfanden wir als optimal. Windows und OS/2 wird bereits jetzt unterstützt, und man kann mit der OS/2-Version angeblich sogar Dynalink-Libraries erzeugen. Dies werden wir in einem späteren Test nachprüfen. Insgesamt hinterläßt Stony Brook den Eindruck eines professionellen Werkzeugs. □

Die Autoren arbeiten bei MTS, Mühlheim.

EGA-Wonder und Turbo-Pascal 4.0

Mit Turbo-Pascal 4.0 kann man den hochauflösenden Modus der EGA-Wonder-Karte (800 × 560 Bildpunkte) nicht direkt ansprechen. Man kann den Modus VGAHi des Grafiktreibers EGAVGA.BGI mit dem MS-DOS-Debugger entsprechend patchen.

Vor Beginn der Änderungen muß man den Wert 1AH zu den Registern CS und DS addieren und vor dem Abspeichern wieder abziehen.

Die Reihenfolge der Eingaben ist in *Bild 1* zu sehen. Die durchgeführten Änderungen muß man, wie *Bild 2* zeigt, mit dem Debugger abspeichern.

Als nächstes sollte man die Tabelle für die VGA mit dem Enter-Befehl von Debug ändern (*Bild 3*). Die alten Werte des VGA-Hi-Modus lauten 27FH für 639 Bildpunkte (maximaler X-Wert) und 1DFH für 479 Bildpunkte (maximaler Y-Wert). Sie werden durch 31FH und 22FH ersetzt. Der Wert 2710H steht für den Verzerrungsfaktor bei Kreisen und braucht nicht geändert zu werden.

In *Bild 4* wird die Neuprogrammierung der Bildpunktberechnung und in *Bild 5* die Änderung von Konstanten im Programmcode gezeigt.

Die Bedeutung der Werte 1DFH (22FH), 50H (64H) und 27FH (31FH) wurden bereits erläutert. 12H steht für den VGA-Modus, der über den Interrupt 10H aktiviert wird. Bei dem hochauflösenden Modus der EGA-Wonder muß man diesen Wert durch 53H ersetzen. Die Größe einer Grafikseite in Paragraphen ist ursprünglich 960H (640 × 480/16/8). Der neue Wert lautet DACH (800 × 560/16/8).

Nach den erfolgten Änderungen wird in einem Pascal-Programm mit GRAPHDRIVER := 9 und GRAPHMODE := 2 der Modus 800 x 560 aufgerufen. Die Grafik-Funktionen können wie gewohnt benutzt werden. Wegen der Neuprogrammierung lassen sich einige der Grafik-Modi des Treibers EGAVGA.BGI jedoch nicht mehr einsetzen. Als gutes Hilfsmittel für die Entwicklung des Patches hat sich das Buch „Turbo-Pascal 4.0 – Konzepte, Analysen, Tips und Tricks" von Arne Schäpers erwiesen.

Otto Donhauser

```
DEBUG EGAVGA.BGI    -RDS
-RCS                DS 2C80
CS 2C80             :2C9A
:2C9A
```

Bild 1. Patchen des Grafiktreibers

```
-RCS                -RDS
CS 2C9A             DS 2C9A
:2C80               :2C80
                    -W
```

Bild 2. Sichern der Änderungen

```
-E0200
2C9A:0200   00.     7F.1F   02.03   DF.2F   01.02   7F.1F   02.03   DF.2F
2C9A:0208   01.02   28.     23.     58.     1B.     10.     27.     08.
```

Bild 3. Änderung der Tabelle für VGA

```
Adresse       alt           neu

2C9A:03FA     SHL  DI,1     PUSH DX
2C9A:03FC     MOV  BX,DI    MOV  AX,DI
2C9A:03FE     MOV  CL,02    MOV  BX,0064    ;($64=100, d.h. 100 Byte/Zeile,
2C9A:0400     SHL  DI,CL    MUL  BX         ; d.s. 100*8=800 Pixels Auflö-
2C9A:0402     ADD  DI,BX    MOV  DI,AX      ; sung in X-Richtung)
2C9A:0404     INC  CL       POP  DX
2C9A:0406     SHL  DI,CL    MOV  CL,03
                            NOP
2C9A:0408     MOV  BX,SI    MOV  BX,SI
```

Bild 4. Die Bildpunktberechnung wird neu programmiert

```
Adresse       alt                              neu

2C9A:0448     MOV  DX,01DF                     MOV  DX,022F
2C9A:044B     MOV  AL,12                       MOV  AL,53
2C9A:044D     MOV  DI,0960                     MOV  DI,0DAC
2C9A:0450     JMP  0455                        JMP  0455
2C9A:0452     MOV  DI,0800                     MOV  DI,0800
2C9A:0455     MOV  WORD PTR [050C],0050        MOV  WORD PTR [050C],0064
2C9A:045B     MOV  [03F1],AL                   MOV  [03F1],AL
2C9A:045E     MOV  CX,027F                     MOV  CX,031F
2C9A:0461     CMP  AL,0D                       CP   AL,0D
```

Bild 5. Auch einige Konstanten muß man ändern

Peter Guthseel

Parallelprogrammierung und Transputer-Netzwerke in C

Ein C-Compiler wird für Transputer erst dann interessant, wenn er Parallelität und Kommunikation der Transputer auszunutzt. Nur eine Programmiersprache, welche die Kommunikationsmöglichkeiten über Kanäle und die Mehrprozeßfähigkeit des Transputers ausnützt, ist adäquat.

Bisher gibt es als reine Parallelsprache für Transputer nur OCCAM, eine vollkommen neu konzipierte Sprache, die alle Möglichkeiten des Transputers ausschöpft, aber nicht alle Features einer modernen Sprache bietet. Es fehlen zum Beispiel Datenstrukturen (außer Arrays) völlig, rekursive Programmierung ist nicht möglich. Es gibt viele Gründe, die für das Beibehalten von herkömmlichen Programmiersprachen sprechen. Zum einen ist es nicht immer einfach sich in eine neue Sprache einzuarbeiten, zum anderen existieren in fast allen Anwendungsfeldern bereits Programme in klassischen Programmiersprachen, die alle neu umgeschrieben werden müßten. Deshalb wird es bald mehrere Compiler für klassische Sprachen geben, die Parallelverarbeitung unterstützen.

Es gibt verschieden Methoden, eine herkömmliche Sprache an die Parallelprogrammierung anzupassen. Eine Methode besteht darin, daß man zusätzliche Kontrollstrukturen in die Sprache einzubaut. Das würde eine Veränderung der Sprachdefinition bedeuten, was die Portabilität von Programmen beeinträchtigen würde. Ein neuer Dialekt würde entstehen und die Sprachverwirrung würde größer werden. Man kann aber auch bei ungeänderter Sprachdefinition spezielle Funktionen und Prozeduren neu hinzufügen, die Parallelprogrammierung möglich machen. Eine Sprache, die sich dafür besonders eignet, ist C. C kann beliebig durch Bibliotheksfunktionen erweitert werden. Die Transputer-C-Version, die hier geschildert wird, bekommt ihre Parallel-Fähigkeit auf diesem zweiten Weg. Das hat sich schon in einigen Applikationen als einfach und praktisch erwiesen.

Parallele Programmierung ...

Wenn nur Bibliotheksfunktionen zugelassen sind, um Parallelität zu ermöglichen, ergeben sich bestimmte Einschränkungen. Die Funktion start_p() erlaubt es zum Beispiel, eine Funktion parallel zum Hauptprogramm ablaufen zu lassen. Mit dieser Funktion ist es möglich C-Funktionen als parallele Prozesse zu starten. Die kleinste Einheit, die parallel abgearbeitet werden kann, ist dabei eine Funktion, da als Parameter nur eine Funktionsadresse übergeben werden kann. Anders als in OCCAM können hier nicht einzelne Ausdrücke parallelisiert werden, da sie sich in C nicht lokalisieren lassen. Allerdings muß diese Einschränkung im Hinblick auf den Aufwand bewertet werden, den es kostet, wenn ein einzelner Ausdruck parallel gestartet werden soll und synchron zu anderen Programmteilen absolviert werden soll. Die Synchronisation zwischen einzelnen Prozessen kann in der hier betrachteten C-Version in die Hände des Anwenders gelegt werden. Dieser kann Prozesse gemeinsame Variablen benutzen (shared memory) oder die Prozesse über Kanäle kommunizieren lassen (message passing). Bei Kanalkommunikation wird die Betriebsmittelvergabe direkt vom Prozessor übernommen. Bei gemeinsamen Variablen ist der Anwender für die gegenseitige Sperrung von Prozessen selbst zuständig.

... und Kommunikation

Für die Kommunikation paralleler Prozesse existieren verschiedene Modelle. Der Transputer ist für das „Message Passing" prädestiniert. Er stellt spezielle Maschinenbefehle für die Kanalkommunikation zur Verfügung. Diese Befehle sind einfach anzuwenden und benötigen bei ihrer Ausführung keine Prozessorzeit, da sie im DMA-Modus auf den Speicher zugreifen. Die Funktionen put_channel() und get_channel() nützen diese Eigenschaft der Hardware der Transputer direkt aus. Sie verwenden entweder eine Linkadresse oder eine Speicheradresse zur Kommunikation. Die Synchronisation von Sender und Empfänger ergibt sich aus der Arbeitsweise der Maschinenbefehle. Ein Prozeß, der put_channel() aufruft, wird solange gestoppt bis ein anderer Prozeß ein korrespondierendes get_channel() aufruft. Wird ein get_channel() zuerst aufgerufen, stoppt der Empfängerprozeß bis ein entsprechender Senderprozeß put_channel() aufruft. Bevor jedoch die Kanalkommunikation verwendet werden kann, muß der entsprechende Kanal, egal ob Link oder Speicheradresse, zurückgesetzt werden. Mit den bisher beschriebenen vier Funktionen kann man schon eine Menge anfangen. In *Bild 1* wird die parallele Programmierung in einem kleinen Beispiel gezeigt. Die Funktion dieses Programms ist sehr einfach. Von den Links 1,2 und 3 werden Integerzahlen eingelesen, addiert und an Link 0 wieder ausgegeben. Man kann sicher auf einfachere Weise Zahlen addieren. Aber an diesem Beispiel kann die Arbeitsweise der bisher vorgestellten Funktionen anschaulich dargestellt werden. Dies sind sozusagen die Grundfunktionen unserer Parallelbibliothek. Um jedoch wirklich alle Möglichkeiten des Transputers ausnutzen zu können, sind noch einige weitere Funktionen implementiert.

Die Timer des Transputers

Der Transputer besitzt für jede Priorität einen Zähler. Die Funktion wait_timer() wartet eine bestimmbare Anzahl von Ticks ab. Der aufrufende Prozeß wird um diese Anzahl Ticks gestoppt. Der Prozessor ist diese Zeit jedoch frei für andere Prozesse. Mit den Funktionen get_timer() und set_timer() lassen sich Zeitmessungen innerhalb von Prozessen durchführen. Diese Zeitmessungen sind aber mit Vorsicht zu betrachten, da es darauf ankommt, wieviele aktive Prozesse sich gerade im Transputer befinden. Nur wenn ein einziger Prozeß aktiv ist, können Zeitunterschiede in echte Zeit umgerechnet werden. Es ist zu beachten, daß die Zähler verschiedener Prioritäten mit unterschiedlicher Taktrate zählen.

Events in C

Der Event-Eingang des Transputers ermöglicht die Erkennung von Interrupts. Diese können auch von C aus erkannt werden. Die Funktion get_event() wartet auf ein Event. Um Interrupts bearbeiten zu können, sollte ein Prozeß mit hoher Priorität gestartet werden, der die Funktion get_event() aufruft. Kehrt diese Funktion zurück, kann der Prozeß im weiteren Verlauf den Interrupt abarbeiten *(Bild 2)*. Ein Prozeß hoher Priorität ist notwendig, damit niedrig priorisierte Prozesse sofort unterbrochen werden können. Außerdem kann ein solcher Prozeß selbst nicht wieder von anderen Prozessen unterbrochen werden. Die Funktion _iniint() initialisiert das System. Tritt ein Event auf, so prüft die Funktion _inter(), ob der Schnittstellenbaustein diesen Interrupt ausgelöst hat. Das empfangene Zeichen wird in einen Ringpuffer gespeichert. Die Funktion getrs232() holt die Zeichen aus dem Puffer ab.

Netzwerke

Bis zu diesem Punkt wurde das Parallelkonzept nur auf einen einzelnen Transputer ausgerichtet. Es ist zwar interessant, wie verschiedene Prozesse auf einem Prozessor ablaufen und sich gegenseitig beeinflussen, aber wirkliche Parallelität erreicht man damit nicht. Echte Parallelität und der damit verbundene Geschwindigkeitsgewinn wird nur in Transputer-Netzwerken erreicht.

Mit intelligenter Programmgestaltung und dem Abwägen zwischen Kommunikations- und Programmgeschwindigkeit läßt sich auf einem Netz mit n Transputern wirklich fast die n-fache Geschwindigkeit gegenüber einem Eintransputersystem erreichen. Der Transputer besitzt vier Link-Anschlüsse mit Übertragungsraten von bis zu 800 KByte/s, über die er mit anderen Transputern vernetzt werden kann. Ein Netzwerk kann verschiedene Formen annehmen. Je nach Anwendungsfall können sich bestimmte Strukturen als besonders geeeignet erweisen. Ein Netzwerk kann beliebig viele Transputer enthalten. Einige Beispiele für Netzwerke werden in *Bild 3* gezeigt.

Die Programmierung von Transputer-Netzwerken wirft gewisse Probleme auf. Es ist davon auszugehen, daß nicht immer bekannt ist, wie ein Netzwerk strukturiert ist und wieviele Transputer es enthält. Um Software erstellen zu können, die auf beliebig komplexen Netzwerken läuft, sollte es eine Möglichkeit geben, dieses unbekannte Netzwerk zu analysieren.

```
/*
  Beispielprogramm 1

  Datenempfang und Verarbeitung von Link 1,2,3 nach Link 0

  Compiler: CCT (CESYS GmbH) mit PARLIB

  Copyright  1988

                    CESYS GmbH

                    Henkestr. 8
               8520 Erlangen

*/

#include <stdio.h>
#include <parlib.h>

/* Variablen für die Prozesskommunikation */

int p1_to_0;
int p2_to_0;
int p3_to_0;

main()
{
        extern get1(),get2(),get3(),put0();

        res_channel(LINKIN1);    /* Link Kanäle zurücksetzen */
        res_channel(LINKIN2);
        res_channel(LINKIN3);
        res_channel(LINKOUT0);

        res_channel(&p1_to_0);  /* Prozesskommunikation zurücksetzen */
        res_channel(&p2_to_0);
        res_channel(&p3_to_0);

        start_p(get1,NORMSTACK,LOWPRI);
        start_p(get2,NORMSTACK,LOWPRI);
        start_p(get3,NORMSTACK,LOWPRI);
        start_p(put0,NORMSTACK,LOWPRI);

}

get1()
{
        int i;
        while (1) {
                get_channel(LINKIN1,&i,sizeof(i));
                put_channel(&p1_to_0,&i,sizeof(i));
        }
}

get2()
{
        int i;
        while (1) {
                get_channel(LINKIN2,&i,sizeof(i));
                put_channel(&p2_to_0,&i,sizeof(i));
        }
}

get3()
{
        int i;
        while (1) {
                get_channel(LINKIN3,&i,sizeof(i));
                put_channel(&p3_to_0,&i,sizeof(i));
        }
}

put0()
{
        int i,i1,i2,i3;
        while (1) {
                get_channel(&p1_to_0,&i1,sizeof(i1));
                get_channel(&p2_to_0,&i2,sizeof(i2));
                get_channel(&p3_to_0,&i3,sizeof(i3));
                i = i1 + i2 + i3;
                put_channel(LINKOUT0,&i,sizeof(i));
        }
}
```

Bild 1. Einen Zahlenstrom über die Links addieren

```
/*

  Beispielprogramm 2

  Es wird ein Interruptgesteuertes gebuffertes get_rs232 implementiert.

  Es wird die Event Logik und die RS 232 Schnittstelle der
  mc-Transputerkarte verwendet.

  Compiler: CCT (CESYS GmbH) mit PARLIB

  Copyright  1988

                    CESYS GmbH

                    Henkestr. 8
               8520 Erlangen

*/

#include <stdio.h>
#include <stddef.h>
#include <parlib.h>

#define STATUS2 0x24
#define RECTRA2 0x2c
#define RXMASK  0x01
#define TXMASK  0x04
#define IMR     0x14
#define MRB     0x20
#define CRB     0x28
#define CSRB    0x24
#define ACR     0x10

#define BUFFLEN 1024                    /* size of i/o-buffer */

#define EVENT           0x20000000L     /* adress of eventlogik */
#define IO_INT_MASK     0x08            /* mask for i/o - interupt */

int     in_index,
        out_index,
        count,
        overflow;

char    iobuff[BUFFLEN];

/* ////////////////////////////////////////////////////////////////////////////
 _inirs2 : initialise rs-port
//////////////////////////////////////////////////////////////////////////// */

_inirs2()
{
 * ((char *) CRB ) = 0x10;
 * ((char *) MRB ) = 0x13;              /* Int on RxReady */
 * ((char *) MRB ) = 0x0f;              /* Int on RxReady */
 * ((char *) IMR ) = 0x20;
 * ((char *) ACR ) = 0xf0;              /* */
 * ((char *) CSRB) = 0xcc;              /* 19200 Baud */
}

/* ////////////////////////////////////////////////////////////////////////////
 _getch2:  gets a char from i/o-port 2
//////////////////////////////////////////////////////////////////////////// */

_getch2()
{
 while (!(*((char *) STATUS2) & RXMASK)) ;
 return *((char *) RECTRA2);
}
```

Bild 2. Mit Interrupts arbeiten

```
/* ////////////////////////////////////////////////////////////////////////////
 _rdychar : checks, if char ready
//////////////////////////////////////////////////////////////////////////// */

_rdychar2()
{
 return ( *((char *) STATUS2) & RXMASK);
}

/* ////////////////////////////////////////////////////////////////////////////
 _iniint : initialise i/o-interuptsystem
           will be called by initio()
//////////////////////////////////////////////////////////////////////////// */

_inibuff()
{
 count=0;
 in_index=0;
 out_index=0;
 overflow=0;
}

_iniint()
{
 _inirs2();
 _inibuff();
 res_channel(EVENT);
 start_p(_inter,NORMSTACK,HIGHPRI);
}

/* ////////////////////////////////////////////////////////////////////////////
 _inter : this routine will be called on interupt
//////////////////////////////////////////////////////////////////////////// */

_inter()
{
 int    stat;
 char   ch;

 while (1) {
        get_event();
        if ((*((char *) EVENT) & IO_INT_MASK )) continue;/* No i/o-interupt */
        if (!_rdychar2()) continue;                      /* no character */
        ch = _getch2();
        if (count++ == BUFFLEN) overflow=TRUE;
        iobuff[in_index]=ch;                             /* put ch in buffer */
        in_index++;
        if (in_index == BUFFLEN) in_index=0;
 }
}

/* ////////////////////////////////////////////////////////////////////////////
 getrs232 : returned a character from input-stream
            overflow = -1
//////////////////////////////////////////////////////////////////////////// */

getrs232()
{
 int temp;

 while ( out_index == in_index );       /* Warte */
 count--;
 temp=iobuff[out_index];
 out_index++;
 if (out_index == BUFFLEN) out_index=0;
 return temp;
}
```

Der Netz-Analysator

Um die Struktur und die Anzahl der Transputer in einem Netzwerk zu erfahren wird ein sogenannter „Netz-Analysator" eingesetzt. Dieser Analysator läuft auf dem Host-Computer. Die Aufgabe eines Host-Computers kann ein PC mit Link-Adapter übernehmen. Er hat die Fähigkeit, das Netzwerk zurückzusetzen und Programme in einen Transputer des Netzwerkes zu laden. Der Netzwerk-Analysator besteht aus einem Steuerprogramm, daß ein Analyseprogramm in den ersten Transputer des Netzes lädt. Dieses Analyseprogramm (auch Wurm genannt) hat die Fähigkeit sich selbst auf den nächsten Transputer zu booten.

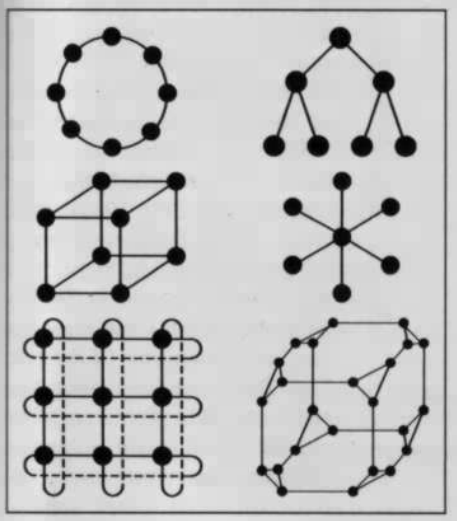


Bild 3. Transputer können vielfältig vernetzt werden

Der Wurm

Nehmen wir mal an, ein Transputer ist mit dem Wurm geladen und ist mit einem anderem Transputer verbunden, in welchem noch kein Wurm läuft. Das erste Wurmprogramm (Vater) sendet sich selbst in den zweiten Transputer (Sohn). Nun kann der zweite Transputer diesen Vorgang, diesmal als Vater, wiederholen. Der Wurm stellt als erstes fest, von welchem Link er gebootet wurde, und schickt über dieses Link die Nachricht, daß er frisch gebootet ist, zurück. Nun testet er die restlichen Links. Befindet sich ein schon gebooteter Transputer an einem Link, empfängt er das Bootprogramm und schickt über dieses Link die Nachricht zurück, daß er schon gebootet war. Ein Link kann also drei Zustände annehmen:

- Wurm wurde geladen
- Wurm war schon geladen
- Timeout

```
Funktionen der Parallelbibliothek

start_p() startet eine C-Funktion ohne Parameter als parallelen Prozeß. Der zu
          reservierende Stackbereich ergibt sich aus den lokalen Variablen der
          Funktion und den lokalen Variablen der aufgerufenen Funktionen. Die
          Priorität kann 0 oder 1 für hohe und niedrige Priorität annehmen.

int start_p(func_adr,stack,pri)
int (*func_adr)();                    /* Startadresse der Funktion */
int stack;                          /* Länge des zu reservierenden Stackbereichs */
int pri;                              /* Prioritätsstufe */

get_pri() gibt die aktuelle Prioritätsstufe zurück.
          Die Priorität kann 0 oder 1 für hohe und niedrige Priorität annehmen.

int get_pri()

-------------------------------------------------------------------------------

Alle Kommunikationsfunktionen können entweder eins der vier Links oder eine
Speicheradresse verwenden.

get_channel() empfängt 'len' Bytes Daten in den Datenbereich 'buffer' von
              Kanal 'channel'

get_channel(channel,buffer,len)
unsigned int channel;           /* Kanaladresse */
char *buffer;                   /* Adresse des Datenbereichs */
int len;                        /* Länge des Datenbereichs */

put_channel() sendet 'len' Bytes Daten aus dem Datenbereich 'buffer' nach
              Kanal 'channel'

put_channel(channel,buffer,len)
unsigned int channel;
char *buffer;
int len;

res_channel() setzt den Kanal mit der Adresse 'channel' zurück. Vor der
              ersten Benutzung eines Kanals muß dieser zurückgesetzt werden.

res_channel(channel)
unsigned int channel;

-------------------------------------------------------------------------------

Alle Timer Funktionen verwenden den Zähler der aktuellen Prioritätsstufe.


wait_timer() wartet bis 'nr' Ticks verstrichen sind.

wait_timer(nr)
int nr;                                 /* Anzahl der zu wartenden Ticks */

get_timer gibt den aktuellen Zählerstand zurück

unsigned int get_timer()

set_timer setzt den Zähler.

set_timer(time)
unsigned int time;
```

Bild 4. Die „Transputer-Bibliotheksfunktionen"

Mit den Daten, die über die Boot Links zurückfließen, läßt sich nun die Struktur des Netzwerkes ermitteln. Das Wurmprogramm beschränkt sich nicht nur auf die Aufgabe des Fortpflanzens, sondern erfüllt auch noch andere Aufgaben. Jeder Transputer bekommt eine eindeutige Nummer zugeordnet und der verfügbare Speicherplatz des Transputers wird berechnet. Diese Informationen fließen über das Netzwerk zurück zum Hauptrechner.

Programmierung des Netzwerkes

Auf dem Hauptrechner kann nun die Netzwerkstruktur ermittelt und gegebenenfalls grafisch dargestellt werden. Die Zuordnung von Prozessen zu Transputern ist nun interaktiv möglich. Die Kenntnis der Struktur des Netzwerkes erlaubt es auch ein Betriebssystem zu laden, mit dem es möglich wird über mehrere Transputer hinweg zu kommunizieren. Jeder Transputer erhält ein Mini-Betriebssystem das folgende Möglichkeiten bietet:

- Zuteilung von Speicher für Prozesse
- Verschicken und Empfangen von Nachrichten
- Weiterleiten von Nachrichten

Dieses Mini-Betriebssystem bietet die Grundlage für beliebig verteilte Anwenderprogramme. □

Günther Sternberg

Low-Cost-CAD-Programm

Beim Schlagwort CAD denkt man meist an leistungsfähige Computer und teure Software. GFA-Systemtechnik möchte mit GFA Draft Plus zeigen, daß man auch für wenig Geld schon ein gutes CAD-Programm bekommen kann. Wir haben dieses Programm einmal für Sie unter die Lupe genommen.

GFA-Systemtechnik in Düsseldorf bietet mit GFA Draft Plus ein Low-Cost-CAD-Programm an. Es soll die PC- und die Atari-ST-Anwender ermuntern, ihre Zeichenbretter in die Ecke zu stellen und den Computer als alternatives Zeichengerät zu nutzen. Auch soll es keineswegs ein einfaches „Malprogramm" für den Anfang sein, das man bereits nach kurzer Zeit gegen ein höherwertiges und damit auch meist teureres Programm austauschen muß. Im Gegenteil, es erhebt den Anspruch, ein vollwertiges CAD-Programm für den professionellen Anwender zu sein.

GFA Draft Plus läuft sowohl am Atari ST als auch am PC unter GEM Desktop, einer grafischen Benutzeroberfläche von Digital Research. Beim Atari ST ist GEM als Standardbetriebssystem bereits im Lieferumfang des Computers enthalten. Der PC-Anwender bekommt das benötigte GEM Desktop als kostenlose Zugabe beim Erwerb von GFA Draft Plus. Damit erhält man für seinen PC neben einem CAD-Programm auch gleich noch eine grafische Benutzeroberfläche.

Weil sowohl die PC-Version als auch die Atari-ST-Version vom GFA Draft Plus auf GEM aufbaut, gibt es, abgesehen von der Installation, keine nennenswerte Unterschiede zwischen den beiden Versionen. Für beide Versionen wird ein gemeinsames Handbuch ausgeliefert, in dem an den betreffenden Stellen zwischen den beiden Computern unterschieden wird. Wir haben daher das Programmpaket nur auf einem Rechner – einem IBM-kompatiblen AT – ausführlich getestet.

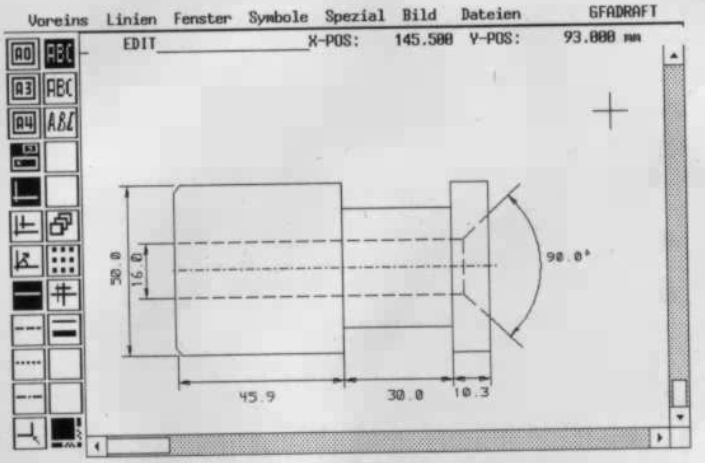


Bild 1. Die Hardcopy wurde mit Hilfe eines im PC-Sonderheft [1] abgedruckten Programms erzeugt

Einfache Installation

Die Atari-Version wird auf zwei 3½-Zoll-Disketten, die PC-Version auf vier 5¼-Zoll-Disketten (360 KByte) ausgeliefert. Hinzu kommen beim PC sechs weitere 5¼-Zoll-Disketten (360 KByte) für GEM Desktop. Die Anforderungen an die Hardware sind in beiden Versionen relativ gering. Beim Atari ST sollte der Hauptspeicher mindestens eine Größe von einem MByte aufweisen, während beim PC zur Not bereits 512 KByte RAM ausreichen. Der PC muß mit zwei Diskettenlaufwerken oder einem Diskettenlaufwerk und einer Festplatte ausgerüstet sein. Als Betriebssystem wird MS-DOS ab Version 2.11 benötigt.

Am PC gilt es als erstes die Grundlage für den Betrieb von GFA Draft Plus zu legen, also GEM zu installieren. Dazu wird die „GEM System Master Disk" ins aktive Laufwerk A eingelegt und die Installation mit „GEMPREP" gestartet. Im Verlauf der Installation muß der Anwender einige Fragen bezüglich seiner Gerätekonfiguration beantworten. Gefragt wird unter anderem nach der eingebauten Grafikkarte, der zur Verfügung stehenden Maus und nach dem angeschlossenen Drucker. Unterstützt werden z. B. LogiMouse, SummaMouse, PC Mouse, alle Mäuse von Microsoft und Grafiktabletts von SummaSketch. Die Angaben für Bildschirmkarte und Maus gelten auch gleich für GFA Draft Plus, während der Drucker nur für die direkte Nutzung von GEM aus maßgeblich ist. GFA Draft Plus verfügt über eigene Programme zur Ausgabe von Zeichnungen auf Plotter und Drucker, die auch getrennt zu installieren sind.

Im Verlauf der GEM-Installation werden die beiden Dateien FORMAT.COM und MODE-.COM gesucht. Diese beiden Dateien, sprich MS-DOS-Befehle, werden von GEM zum Formatieren neuer Disketten bzw. zum Einstellen der RS232-Parameter benötigt. Besitzt nun eine der Dateien die Extension .EXE, so empfiehlt das Handbuch, die Datei einfach umzubenennen. Ob diese Vorgehensweise völlig hundertprozentig ist, erscheint mir fraglich.

Bei einem PC mit Festplatte folgt als nächstes die eigentliche Installation von GFA Draft Plus. Die ganze Installation besteht aus dem Kopieren aller vier Disketten in ein Unterverzeichnis der Festplatte. Im Handbuch wird dieser Kopiervorgang mittels GEM erledigt. Da sich die dazu nötigen Arbeitsgänge bei einem grafischen Betriebssystem nur schwerlich beschreiben lassen, dazu nur soviel, daß die Installation relativ einfach und im Handbuch ganz brauchbar beschrieben ist.

Drucker und Plotter

Um das Thema Installation abzuschließen, auch gleich noch die Festlegung von Drukker und Plotter für die beiden Ausgabeprogramme von GFA Draft Plus. Die Treiber für die gängigsten Drucker und Plotter sind jeweils in einem Unterinhaltsverzeichnis zusammengefaßt. Ist ein Treiber für das eigene Gerät vorhanden, muß man lediglich die entsprechende Treiberdatei unter dem Namen PRINTER.INS bzw. PLOTTER.-INS in das Unterinhaltsverzeichnis mit den Ausgabeprogrammen kopieren. Existiert kein Treiber für das eigene Gerät, kann man sich seinen eigenen Treiber anlegen. Jeder Treiber besteht aus einer Textdatei, in der jeder Zeile eine bestimmte Bedeutung zukommt. Das Konzept der Treiber ist

sehr einfach und trotzdem lobenswert, da man auf diese Weise auch ein ausgefalleneres Gerät selbst installieren kann. So läßt sich beispielsweise ein Kyocera-Laserdrukker mit der internen Beschreibungssprache Prescripe als Plotter betreiben. Der einzige Haken bei der Sache: Der Aufbau und die Erstellung einer solchen Treiberdatei wird im Handbuch nur sehr unzureichend beschrieben. Ein Anfänger dürfte mit dieser Beschreibung kaum etwas anzufangen wissen. Überhaupt ist das Handbuch in bezug auf die Installation und die Einführung in GEM sehr spartanisch ausgefallen.

Ohne Maus geht nichts

Wenden wir uns nun dem Aufruf von GFA Draft Plus zu. Zum Start des Programms muß man sich zunächst in GEM bis zum Unterinhaltsverzeichnis von GFA Draft Plus vorarbeiten. Unter GEM ausführbare Programme tragen die Extension .APP für Applikation. Zum Start des Programms setzt man den Cursor auf das Symbol für die Datei GFADRAFT.APP und betätigt zweimal kurzhintereinander die linke Maustaste. Nach dem Laden von GFA Draft Plus wird der Anwender als erstes aufgefordert den Namen einer bestehenden oder neuen Zeichnung einzugeben. Die Auswahl einer bereits vorhandenen Zeichnung kann auch per Maus erfolgen. Für die ersten Gehversuche kann man eine der Beispielzeichnungen laden. *Bild 1* zeigt eine Hardcopy mit dem Beispiel für die Bemaßung im hochauflösenden Modus einer EGA-Karte. Die grafischen Symbole am linken Rand sind Zustandsschalter und zugleich Zustandsanzeige für die wichtigsten bzw. am häufigsten benötigten Funktionen. So läßt sich durch Anklicken des entsprechenden Symbols per Maus das Hilfsraster ein- und ausschalten oder zwischen der Anzeige einer oder aller Zeichnungsebenen wählen.

Die „Schieber" am unteren bzw. rechten Rand ermöglichen eine Verschiebung des Bildausschnitts. Dazu muß man den Schieber nur mit der Maus „anfassen" und mit gedrückter Maustaste zur gewünschten Stelle bewegen. Zur Vergrößerung der Zeichenfläche läßt sich die Anzeige der „Schieber" auch unterdrücken.

Jedes der Wörter in der obersten Zeile des Bildschirms steht stellvertretend für ein sogenanntes Pull-Down-Menü. Ein Menü wird sichtbar oder wie man sagt, es „klappt" herunter, sobald der Cursor zu dem betreffenden Wort bewegt wird. Eine Aufstellung aller Pull-Down-Menüs ist *Bild 2* zu entnehmen. Die Aktivierung einer Menüfunktion erfolgt durch Anklicken der entsprechenden Zeile mit der Maus.

Leider gibt es keine Möglichkeit, eine Funktion auf andere Weise aufzurufen. Sogenannte „Shortcuts", mit denen ein geübter Anwender mit einem Tastatur-Kommando die Auswahlprozedur abkürzen kann, kennt das Programm leider nicht. Einem Anfänger oder einem Anwender, der das Programm nur gelegentlich benutzt, kommt das Arbeiten mit der Maus sicherlich entgegen. Ein eingearbeiteter Benutzer wird es doch manchmal leid sein, ständig mit der Maus zwischen Menüs und Zeichenfläche hin- und herfahren zu müssen. Er würde oft eine Funktion lieber per Tastatur aufrufen und mit dem Cursor bei der Zeichnung verbleiben.

Zeichenfläche am Bildschirm ist wegen der geringen Auflösung nur in den wenigsten Fällen sinnvoll. Meist holt man sich nur einen kleinen Ausschnitt der Zeichnung auf den Bildschirm. Mit der Funktion „Bildausschnitt-Zoom" des Pull-Down-Menüs „Bild" läßt sich für den Bildausschnitt ein Maßstab von 1:8, 1:4, 1:2, 1:1, 2:1, 4:1 oder 8:1 einstellen. Der Bildausschnitt wird mit Hilfe der „Schieber" am unteren und rechten Rand des Bildschirms über die Zeichnung bewegt. Leider ist mir aufgefallen, daß die Darstellung der Zeichnung am Bildschirm nicht maßstabsgetreu erfolgt. Ein Quadrat mit einer Kantenlänge von 4 cm, laut der angezeigten Koordinaten, erscheint auf

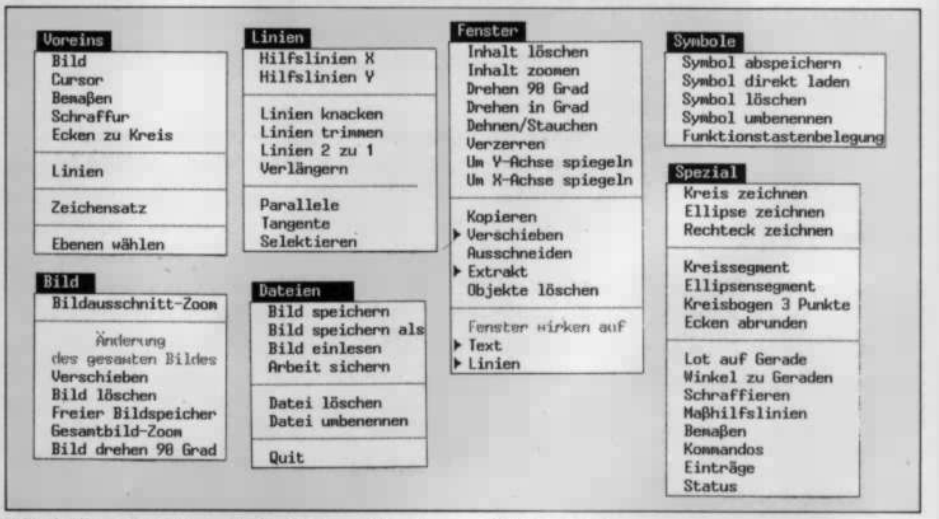


Bild 2. Übersicht über alle Pull-Down-Menüs von GFA Draft Plus

Auch die Festlegung von Programm-Parametern per Dialogbox ist sicherlich nicht jedermanns Geschmack, liegt aber im Trend der Zeit zu grafischen Benutzeroberflächen.

Die Leistungsdaten von GFA Draft Plus sind beeindruckend

Das Programm kennt als Zeichnungselemente Linien, Kreise, Kreissegmente, Ellipsen, Ellipsensegmente, Rechtecke und Texte. Aus diesen Grundelementen baut sich jede Zeichnung auf. Die maximale Größe einer Zeichnung ist von der Größe des Hauptspeichers abhängig, da die Daten einer Zeichnung während der Bearbeitung dort abgelegt werden. Eine Funktion zur automatischen Sicherung der Zeichnung in bestimmten Zeitabständen gibt es leider nicht. Mit einer solchen Funktion läßt sich der Schaden bei einem Stromausfall oder einem Systemabsturz – in diesem Fall geht der Speicherinhalt und damit die Zeichnung ja verloren – in Grenzen halten.

Die maximale Größe der Zeichenfläche beträgt DIN A0. Die Abbildung der gesamten dem Bildschirm bei einer 1:1 Abbildung als Rechteck mit abweichenden Kantenlängen. Das Programm läßt die gleichzeitige Bearbeitung von maximal 10 Ebenen zu. Jede Ebene verkörpert eine eigene Zeichnung und wird in einer eigenen Datei gespeichert. Mehrere Ebenen verwendet man normalerweise nur bei Zeichnungen, bei denen es nötig ist, Teile der Zeichnung getrennt auszugeben oder zu bearbeiten. So nutzt man die Ebenentechnik bei Layouts von Leiterplatten. Auf eine Ebene zeichnet man alle Verbindungen der Lötseite, auf eine weitere Ebene die Verbindungen der Bauteileseite und eine dritte Ebene enthält den Bestückungsaufdruck. Bei GFA Draft Plus kommt den Ebenen zwangsweise noch eine zusätzliche Bedeutung zu. Sobald man eine Zeichnung farbig gestalten möchte, muß man sich der Ebenen bedienen. Jede Ebene läßt sich in einer von 10 möglichen Farben, einen Farbbildschirm natürlich vorausgesetzt, anzeigen. Innerhalb einer Ebene gibt es keine Möglichkeit verschiedene Farben zu nutzen. Diese Vorgehensweise ist umständlich und lästig, da man beispielsweise bei einer Zeichnung mit fünf

Architektur

WASCHB	TUER_80	DOP_TUER	HERD	WC	BETT
DOP_BETT	HOCKER	SCHR_150	SCHR_200	SPUELE	STUHL
TISCH1	TISCH2	TISCH3	TREPPE_1	TREPPE_2	WANNE
DUSCHE					

Elektrotechnik

AMPERM	LAST_SCH	LAST_TR	3POL_UEB	SCHL_HD	SCHUETZ
WECHSLER	3POL_SCH	DR_MOTOR	LEUCHTE	OEFFNER	SICHERG
SPULE	ZAEHLER				

Elektronik

BATTERIE	RAMBL64	SAB2732	SAB8255	AND	ARBKONT
CONDENSA	DIODE	DRIVER	ELKO	E_POTI	IC16
INVERTER	NAND	NFET	NOR	NPN	OP_AMP
OR	PFET	PNP	POTI	QUARZ	RELAIS
RELAIS1	RESIST	RUHEKONT	TRAFO	WECHSLER	XNOR
XOR					

Flußdiagramme

IF_THEN	PROG_MOD	DATENTR	BEGIN	SPEICH_3	PFEIL
CONNECT	END	IN_OUT	MAGNETB	MANUELL	OPERAT
SCHRIFT	SPEICH_1	SPEICH_2	SUBPROC		

Layout

DIP_20	DIP_18	DIP_16	DIP_14	DIP_40	QUAD_68
HOLE_10	RAMLA	BP_IC_16	BP_COND	DIP_24	DIP_28
DIP_06	DIP_08	HOLE_08	HOLE_15		

Beschriftungsrahmen

SCHRFELD	A3H	DINA4Q	A3Q	A4H	DINA4H
DINA3Q	A4Q	DINA3H			

Bild 3. Die mitgelieferten Symbole sind für Übungszwecke geeignet

Farben immer fünf Ebenen bearbeiten und somit auch fünf Dateien mitschleppen muß. Das Programm ist also für farbige Zeichnungen nur bedingt geeignet.

Praktisch: Symbole

Die größte Arbeitserleichterung und den größten Zeitgewinn bei der Nutzung eines CAD-Programms resultiert aus dem Umstand, daß man ein häufig benötigtes Teil einer Zeichnung, zum Beispiel das Schaltzeichen eines Widerstands, als sogenanntes Symbol abspeichern kann. Damit kann ein einmal gezeichnetes und abgespeichertes Symbol bei Bedarf schnell geladen und an der gewünschten Stelle in die Zeichnung übernommen werden. Das Arbeiten mit Symbolen ist bei GFA Draft Plus sehr einfach. Ein vorhandenes Symbol wird mit der Funktion „Symbol direkt laden" des Menüs „Symbole" geladen. Das Symbol folgt bei gedrückter linker Maustaste jeder Bewegung des Cursors. Durch Betätigung der linken Maustaste wird das Symbol an der aktuellen Position fixiert, während eine Betätigung der rechten Maustaste das Symbol wieder verschwinden läßt. Die zehn am häufigsten benötigten Symbole lassen sich jeweils einer der zehn Funktionstasten zuordnen. Durch Drücken der entsprechenden Funktionstaste wird das Symbol geladen und läßt sich sofort plazieren. Die Erstellung eigener Symbole ist denkbar einfach. Der als Symbol abzuspeichernde Teil einer Zeichnung wird durch ein rechteckiges Fenster markiert und dann die Funktion „Symbol abspeichern" aufgerufen.

Beim Kauf von GFA Draft Plus werden derzeit einige fertige Symbole (*Bild 3*) aus verschiedenen Anwendungsbereichen mitgeliefert. Diese Symbole entsprechen meist nicht den einschlägigen Normen und sind mehr zur Übung gedacht. Umfangreichere Bibliotheken mit Möbel-Symbolen und für die Bereiche Elektrotechnik und Pneumatik-Hydraulik sind für jeweils 99 DM erhältlich.

GEM bremst CAD aus

Die Einbindung des CAD-Programms in die grafische Benutzeroberfläche GEM bringt Vor- und Nachteile. Der größte Vorteil für den Anwender von GFA Draft Plus ergibt sich aus der einfachen und leicht zu erlernenden Benutzeroberfläche. Auch ein Anfänger kann sich innerhalb weniger Stunden in GFA Draft Plus einarbeiten. Die grafische Oberfläche in Form der Pull-Down-Menüs, der „Schieber", der Icon-Schalter, der OK- und Abbruch-Schalter und der Dialogboxen verkürzt die Einarbeitungszeit und erleichtert das Arbeiten.

Auf der anderen Seite geht ein gewisser Teil der Rechnerleistung für die Erledigung interner Aufgaben von GEM verloren. Dies äußert sich auch noch bei einem 6-MHz-AT in Form eines „träge" reagierenden und etwas sprunghaften Cursors sowie einer gewissen Verzögerungszeit bei der Betätigung einer Maustaste. Beim Verlegen eines Linienzugs führt jede Betätigung der linken Maustaste zum Setzen eines neuen Eckpunkts an der aktuellen Cursorposition. Zwischen der Betätigung der Maustaste und dem wirklichen Plazieren des Eckpunkts vergeht aber ungefähr eine halbe Sekunde, in der die Maus nicht bewegt werden darf. Ansonsten landet der Eckpunkt nicht an der geplanten Stelle. Auch die Geschwindigkeit beim Aufbau des Bildes dürfte ohne GEM um einiges höher sein.

Kein vollständiger Ersatz für Profi-CAD-Programme

GFA Draft Plus ist auf der aktuellen Entwicklungsstufe durch das Fehlen bestimmter Funktionen, wie zum Beispiel Splines und Hilfen für räumliches Konstruieren, keine Konkurrenz für die „großen" CAD-Programme. Die meisten Anwender stellen jedoch wesentlich geringere Anforderungen an ein CAD-Programm. Für diese Gruppe ist das Programm eine Alternative zu oft erheblich teureren Produkten. Ein paar kleine Kinderkrankheiten in der ersten Version sind noch zu beseitigen. So landeten Druck- und Plotausgaben für die Centronics-Schnittstelle immer auf der seriellen Schnittstelle, worüber sich die dort angeschlossene Maus ganz und gar nicht erfreut zeigte. Solche kleinen und wohl auch bald nachgebesserten Fehler konnten den ansonsten guten Eindruck von GFA Draft Plus nicht schmälern. Der gute Eindruck beruht nicht nur auf dem guten Preis-/Leistungsverhältnis, er entsteht vor allem, weil das Programm wirklich semi-professionelles Arbeiten zuläßt.

Literatur

[1] *Frank Brendle*: Das PC-Sonderheft. Franzis-Verlag, Sonderheft Nr. 238, Seite 32.

Ingo Eickmann, Bernd Ewert

Potenzieren unter Turbo-Pascal

Jeder kennt das Problem: Hochsprachen wie Turbo-Pascal enthalten einige mathematische Funktionen als Befehle; viele müssen jedoch aus den vorhandenen erst gebildet werden. Dieser Artikel beschreibt die Implementation eigener Funktionen am Beispiel der Potenz y^x in Turbo-Pascal 3.0

Die Einbindung läßt sich leicht an andere Sprachen anpassen.

Die Berechnung von y^x kann selbstverständlich in

$$y^x = e^{x * \ln y} = 10^{x * \log y} = 2^{x * \mathrm{ld}\, y}$$

umgeformt werden, da wir uns bei beliebigen Exponenten x auf eine positive Basis y beschränken wollen. Dies läßt sich so auch mit den vorhandenen Pascalbefehlen berechnen, allerdings mit Genauigkeitseinbußen, wenn kein 8087 vorhanden ist, bzw. unterstützt wird. Sobald die Berechnungszeit wichtig wird, ist aber eine kurze Assemblerroutine notwendig. Um Nägel mit Köpfen zu machen, wird der 8087 und die Turbo-Pascal-Version mit 8087-Unterstützung verwendet.

Programmbeschreibung

Da sich auf dem 8087 die Berechnungen zur Basis 2 schnell realisieren lassen, wird die Potenz in

$$y^x = 2^{x * \mathrm{ld}\, y}$$

umgeformt. Die dazugehörigen Programmlistings sind in *Bild 1* und *Bild 2* abgedruckt, das Flußdiagramm der Berechnung in *Bild 3*. Nach den bei der Einbindung von Assemblerroutinen in Turbo-Pascal üblichen Befehlen zur Sicherung der kritischen Register, reserviert das Programm auf dem Stack einen Bereich für eigene Anwendungen. Hier werden später die Flags zur Steuerung der Potenzberechnung und der Coprozessor-Status abgelegt. Der für die Berechnung notwendige Wert 0,5 wird dort als Short-Real der Länge 4 Byte (00 00 00 3F) in der Reihenfolge LSB bis MSB gespeichert.

Nach der Initialisierung des 8087 werden der Exponent und die Basis eingelesen. Die Basis wird auf 0 überprüft. Hierzu muß nach dem Vergleich (FCOMP) das Statuswort gelesen werden. Das Zero-Flag (Bit 6) kann mit einem TEST-Befehl überprüft werden, eleganter ist jedoch, die obere Hälfte des Statuswortes in die untere Hälfte des Flagwortes des 8086/88 einzulesen und einen bedingten Sprung auszuführen. Dies ist möglich, da das Carry-, das Parity- und das Zero-Flag in diesen Bytes die gleiche Position einnehmen.

Nun kann in die eigentliche Berechnung eingestiegen werden: Für den Fall, daß aus Versehen eine negative Basis vorliegt, was bei der Berechnung des Logarithmus zu Problemen führt, wird der Betrag der Basis gebildet. Da mit den 8087-Potenzierbefehlen nicht beliebige Potenzen berechnet werden können, spalten wir den auf die Basis 2 umgerechneten Exponenten z in drei Teile auf:

$$z = x * \mathrm{ld}\, |y| = u + v + w$$

u ganzzahliger Anteil
v 0,5-Anteil (0 oder 0,5)
w eine Zahl kleiner 0,5

Bei negativem Exponenten wird das Flag 2 gesetzt, was am Ende der Potenzberechnung die Bildung des Kehrwertes bewirkt. Durch setzen des Round Control auf Chop (RC=11), d. h. Abschneiden in Richtung 0 bei Wechsel des Darstellungsformats, erzeugt der Befehl FRNDINT den ganzzahligen Anteil des Exponenten.

Wer Wert auf besondere Genauigkeit legt, kann Round Control hiernach wieder auf Round to Nearest (RC=00) setzen. Der Befehl F2XM1 führt die Berechnung von $2^w - 1$ aus. Zur Korrektur wird die 1 direkt wieder addiert. Für den Fall, daß V = 0,5 war, wird das Ergebnis E mit $2^{0,5}$ multipliziert. Der ganzzahlige Anteil u wird mit FSCALE in die Berechnung einbezogen ($E = E * 2^u$).

Nach der bei gesetztem Flag 1 durchgeführten Kehrwertbildung wird das Resultat direkt an der vorher für das Funktionsergebnis festgelegten Stelle ES:DI abgelegt. Nach Wiederherstellung des Stacks und der Register erfolgt der Rücksprung zum aufrufenden Programm. Anmerkung: Der mathematisch unbestimmte Ausdruck 0^0 wird in der Routine so behandelt, daß das Ergebnis 0 ist.

Kurzbeschreibung der verwendeten 8087-Befehle

Befehl	Einfluß auf Stack	Operation
FABS	–	Bildet den Betrag der Zahl im Stack-Top
FADD	ST=ST+1 nach OP	ST(1)=ST(1)+ST
FDIV	ST=ST+1 nach OP	ST(1)=ST(1)/ST
FMUL	ST=ST+1 nach OP	ST(1)=ST(1)*ST
FSUB	ST=ST+1 nach OP	ST(1)=ST(1)–ST
FSQRT	–	ST=sqrt(ST) (Wurzel ziehen)
FYL2X	ST=ST+1	Z=Y*ld X (X in ST, Y in ST(1), Z in ST(1))
F2XM1	–	$Y=2^x-1$ (X in ST, Y in ST)
FSCALE	–	$Z=Y*2^x$ (X in ST(1), Y in ST, Z in ST)
FXCH	–	Vertauschen von ST mit ST(1)
FRNDINT	–	Umwandlung von ST in Integer
FINIT	SP wird gelöscht	Initialisierung des 8087

```
;-------------------------------------------------------------------
; Berechnung einer beliebigen Potenz zu einer positiven Basis
;
; (C)88 Ingo Eickmann u. Bernd Ewert
;-------------------------------------------------------------------

code segment
        assume cs:code

YHOCHX proc near

start:  push bp
        mov bp,sp
        push si
        push ds
        push cs
        pop ds
        lea di,[bp+14h]             ; Zeiger auf Funktionsergebnis
        push ss                     ; in ES:DI  = SS:[BP+14h]
        pop es

        mov ax,3f00h                ; Konstante 0.5 als double Word
        push ax                     ; auf Stack schreiben
        xor ax,ax
        push ax                     ; weitere 4 Bytes auf Stack
        push ax                     ; reservieren
        push ax
        mov si,sp
        finit                       ; 8087 initialisieren

        fld qword ptr [bp+4]        ; Exponent einlesen
        fld qword ptr [bp+0ch]      ; Basis einlesen
        fldz
        fcomp st(1)
        fstsw word ptr es:[si+2]    ; Statuswort auf Stack speichern
        fwait
        mov ah,byte ptr es:[si+3]
        sahf                        ; Flags mit Condition Code laden
        je  raus                    ; wenn Basis = 0 dann Ergebnis = 0

        call Berechnung

raus:   fstp qword ptr es:[di]      ; Endergebnis übergeben
        add sp,8                    ; Stack von lokalen Variablen befreien
        pop ds
        pop si
        mov sp,bp
        pop bp
        ret 10H                     ; Stack freigeben bis zum Funktions-
                                    ; ergebnis: Zwei Realzahlen hatten
                                    ; 16 = 10h Byte belegt

YHOCHX  endp

BERECHNUNG proc near
        fabs                        ; z=ld |y|
        fyl2x                       ;
        code ends
        end
```

```
        fldz                        ; neg. Exponent ?
        fcomp st(1)                 ; wenn ja, Flag 1 ES:[SI] setzen
        fstsw word ptr es:[si+2]
        fwait
        mov ah,byte ptr es:[si+3]
        sahf
        jb gr0
        mov al,1
        mov byte ptr es:[si],al

gr0:    fabs                        ; Betrag der Basis bilden
        fstcw word ptr es:[si+2]    ; Round Control auf Chop setzen
        fwait
        mov bx,0C00H
        mov ax,word ptr es:[si+2]
        or ax,bx
        mov word ptr es:[si+2],ax
        fldcw word ptr es:[si+2]

        fld st(0)                   ; Integer des Exponenten bilden
        frndint                     ; und vom ursprünglichen Exponenten
        fsub st(1),st               ; subtrahieren
        fxch st(1)

        fld dword ptr es:[si+4]     ; Exponent auf [0;0,5> normieren und
        fcom st(1)                  ; Flag 2 setzen wenn Exponent größer
        fstsw word ptr es:[si+2]    ; als 0.5 war
        fwait
        mov ah,byte ptr es:[si+3]
        sahf
        jnb gr05
        fsub st(1),st
        mov al,1
        mov byte ptr es:[si+1],al
gr05:   fstp st(0)

        f2xm1                       ; 2^Exponent berechnen und ggf.
        fld1                        ; mit Wurzel von 2 multiplizieren
        faddp st(1),st
        fwait
        mov al,byte ptr es:[si+1]
        or al,al
        jz keinf12
        fld1
        fadd st(0),st
        fsqrt
        fmulp st(1),st

keinf12: fscale                     ; 2^Exponent * 2^Integer

        mov al,byte ptr es:[si]     ; bei neg. Exponent : Ergebnis^(-1)
        or al,al
        jz zurueck
        fld1
        fxch st(1)
        fdivp st(1),st
zurueck: ret

BERECHNUNG endp
```

Bild 1. Das 8086-Assemblerprogramm zur Berechnung beliebiger Potenzen

```
{
 ------------------------------------------------------------------------
  Programm zur Berechnung einer beliebigen Potenz

  (C)88 Ingo Eickmann u. Bernd Ewert
 ------------------------------------------------------------------------
}

VAR  b,e,c : real;

FUNCTION y_hoch_x ( basis,exponent : real ):real ; external 'yhochx.com';


BEGIN {Hauptprogramm}

  clrscr;
  write('Basis     :' );readln(b);
  write('Exponent  :' );readln(e);
  c := y_hoch_x (b,e);
  writeln ('Ergebnis  : ',c);

END. {Hauptprogramm}
```

Bild 2. Verwendung der Potenzberechnung aus einem Turbo-Pascal-Programm heraus

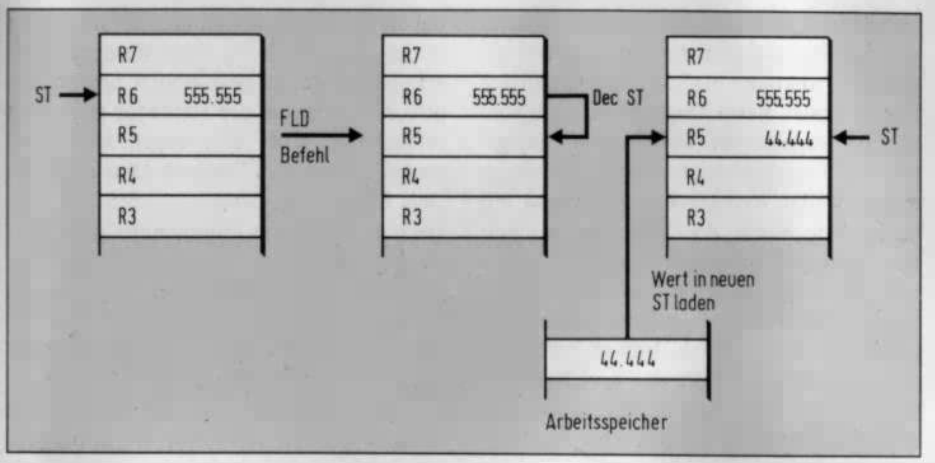


Bild 4. Schema der Arbeitsschritte eines Ladebefehls

Verwendete 8087-Befehle

Zur Erleichterung des Programmverständnisses sind im Folgenden die benutzten 8087-Befehle in ihrer Wirkungsweise zusammengestellt.

Ladebefehle
Konstanten, Speichervariablen oder Inhalte von 8087-Stack-Registern können in den 8087 eingelesen werden. Die Ladebefehle beziehen sich immer auf ein Register, welches als sogenannter Top-of-Stack (ST oder ST(0)) gekennzeichnet ist. Der Zeiger auf ST wird vor dem Laden des entsprechenden Parameters dekrementiert. Auf diese Weise können die 8 Register des 8087 mit Werten belegt werden (*Bild 4*).

Verwendete Ladebefehle:
FLD QWORD PTR [BP+4] ; lädt eine 8 Byte Realzahl
FLDZ ; lädt 0 in ST
FLD1 ; lädt 1 in ST
FLD ST(0) ; Kopiert ein Register in ein anderes

Speicherbefehle
Der Inhalt des Stacktop-Registers wird entweder in ein anderes 8087-Register oder in den Arbeitsspeicher geladen. Die Position des Stacktop wird hierbei nicht verändert. Bei fast allen Speicherbefehlen und mathematischen Operationen kann ein „p" angehängt werden, was ein Inkrementieren des Zeigers auf ST nach dem Ausführen des Befehls zur Folge hat.

Steuerbefehle
Der 8087-Coprozessor stellt Vergleichsbefehle (FCOM) zur Verfügung, welche das Ergebnis einer Vergleichsoperation im Condition-Code als Teil des Statuswortes (Bit 8,9,10,14) ablegen. Nach Speicherung des Statuswortes in den Hauptspeicher kann der Condition-Code zur Kontrolle des Rechenganges ausgewertet werden.
FCOM ST(1) ; Vergleicht den Stacktop mit dem Register ST(1) und schreibt das Ergebnis in das Statuswort.
FSTSW WORD PTR [SI+2] ; Schreibt das Statuswort an die durch SI+2 adressierte Stelle im Arbeitsspeicher.

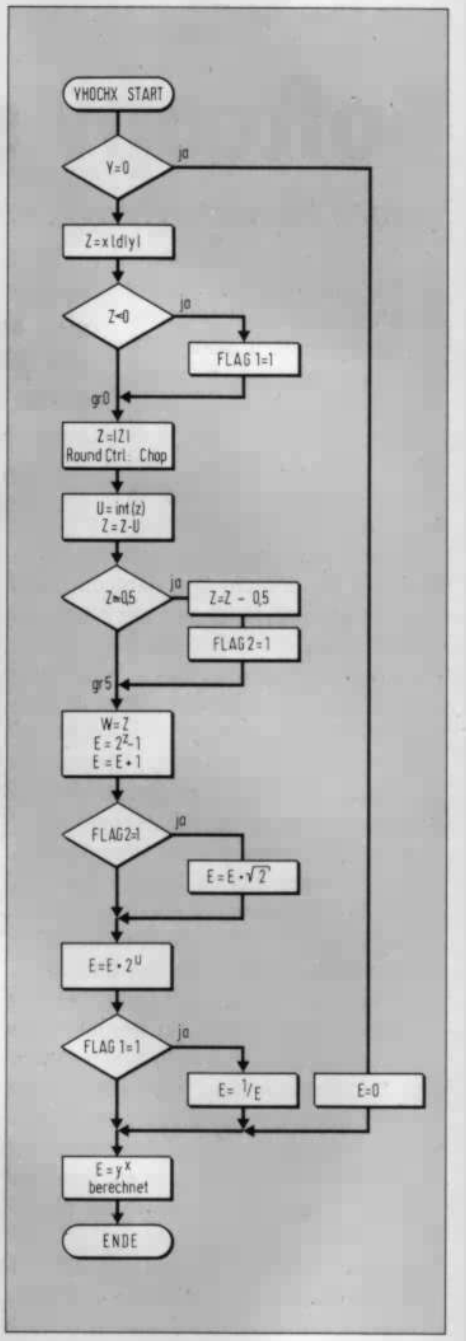


Bild 3. Flußdiagramm zum Assembler-Programm

Zum genauen Aufbau des Status- und Kontrollwortes sei an weiterführende Literatur verwiesen. Eine Kurzbeschreibung der sonstigen verwendeten mathematischen Operationen gibt die Tabelle wieder.

Literatur

[1] Handbuch zu Turbo Pascal Vers. 3.0, Heimsoeth Software
[2] Die 8087/80287-numerischen-Prozessor-Erweiterungen für 8086/80286-Systeme, te-wi Verlag
[3] Programmer's Pocket Reference Guide Intel Corporation

Erich Ludwig

Softcopy statt Hardcopy

Neues Hardcopyprogramm für den Atari ST mit 24-Nadel-Drucker

Immer mehr Anwender des Atari ST entscheiden sich bei der Auswahl des Druckers für ein Modell mit 24 Nadeln. Sie wollen ihre schönen Grafiken gerne optimal zu Papier bringen. Bei einer Hardcopy bringen die Drucker aber nicht, was sie zu leisten imstande wären.

Mit dem Programm, das wir hier vorstellen, wird das Leistungsvermögen vieler 24-Nadel-Drucker voll ausgeschöpft. Ein Vertreter dieser Spezies, der NEC P6 z. B. schafft eine Auflösung von 2000 x 1200 Punkten auf einem DIN A4-Blatt und übersteigt die Auflösung des Atari ST damit wesentlich. Die Hardcopy-Funktion des Atari ST nutzt also nur einen Teil der Auflösung des Druckers.

Für alle Ausgaben, die bildschirmorientiert sind, wie z. B. eine Hardcopy oder der Ausdruck eines Grafikprogrammes, konnte die hohe Auflösung der 24-Nadel-Drucker bisher keinen Vorteil bieten; außer daß das Druckformat bis auf Briefmarkengröße verkleinert werden konnte. Die störenden Stufen in schrägen Linien blieben auch auf einem 24-Nadel-Drucker erhalten (*Bild 1*).

Dieses Problem scheint auch nicht lösbar. Wenn nur die Information des Bildschirmspeichers zur Verfügung steht, können nicht mehr Punkte gedruckt werden, als in ihm definiert sind. Und doch haben diejenigen Bildschirmpunkte, die eine solche Treppe bilden, eine ganz besondere Umgebung. Analysiert man nämlich die Pixel, die jeden Bildschirmpunkt umgeben, bekommt man die nötigen Informationen für die höhere Auflösung des Druckers.

Soll der Bildschirminhalt auf ein DIN A4-Blatt gedruckt werden, stellt der 24-Nadel-Drucker jeden gesetzten Punkt mit 3 x 3 Nadelanschlägen dar. Die weichen Konturen werden dadurch erzeugt, daß bei gesetzten Punkten einzelne Nadelanschläge in den Ecken dieser Quadrate je nach Umgebung weggelassen werden oder bei gelöschten Punkten einzelne Nadelschläge an diesen Stellen dazukommen. Schrägen mit einem Winkel von 45 Grad werden damit ganz stufenfrei dargestellt (*Bild 2*).

Der Beurteilungsalgorithmus für die vier Ecken eines jeden Punktes und seiner Umgebung ist sehr einfach. Immer, wenn bei einem gesetzten Punkt alle drei Nachbarpunkte, die diese Ecke berühren, gelöscht sind, wird die Ecke des betrachteten Punktes gelöscht, sonst bleibt sie gesetzt. Bei der Ecke eines gelöschten Punktes reicht es aus, daß die beiden nichtdiagonalen Nachbarpunkte, die diese Ecke berühren, gesetzt sind. Dann wird an dieser Ecke ein Nadelanschlag hinzugefügt, das heißt, ein Neuntel des Pixels wird gedruckt, obwohl es eigentlich gelöscht ist. Wenn jetzt noch der Sonderfall berücksichtigt wird, daß isoliert stehende Punkte, egal ob gesetzt oder nicht, unverändert bleiben sollen, stehen sämtliche Zusatzinformationen zur Verfügung, um die Hardcopy mit weniger stufiger Erscheinung aufs Papier zu bringen.

Das Programm (*Bild 3*) wurde für den Atari ST geschrieben, der Ausdruck erfolgt auf dem NEC P6/P7. An andere 24-Nadel-Drucker kann das Programm angepaßt werden, wobei die ausführliche Kommentierung helfen dürfte. Die Erstellung einer Hardcopy dauert nicht länger als gewöhnlich, obwohl fast 800 000 Abfragen pro Hardcopy zusätzlich erforderlich sind. Das abgedruckte Programm ist mit dem Profimat-Assembler von Data Becker geschrieben worden.

Nach dem Start des Programmes vom Desktop aus oder gleich beim Booten des Computers aus dem Auto-Ordner heraus, steht die Routine resident im Speicher. Das Programm ersetzt die im Betriebssystem eingebundene Hardcopy-Routine und wird genau wie diese bedient. Sollte der Drucker aus Versehen nicht betriebsbereit sein, wird der Befehl einfach ignoriert.

Probleme kann es in den Situationen geben, wo auch die Hardcopy-Routine des Betriebssystems fehlerhaft arbeitet. Das kann der Fall sein, wenn ein Programm auf einen Tastendruck wartet, wie es beim Auflisten von Dateien der Fall ist. Das Betriebssystem schafft dann manchmal nicht die Zuordnung des Tastendrucks als Hardcopy-Anforderung oder als Signal, die nächste Zeile aufzulisten. □

Bild 1. Einige Ausdrucke der normalen Hardcopy-Routine

Bild 2. Diese Ausdrucke wurden mit der Softcopy-Routine erstellt

```
;                              Softcopy

;Bilschirmausdruck auf dem NEC P6/P7 mit weichen Konturen.
;                    von
;                    Erich Ludwig
;                    Pfaelzer Weg 2
;                    4790 Paderborn
;                    Rueckfragen:  05251/59252

;Parameter:
;Profimat, Assemblierung PC-relativ, relozierbar
;Editorspeicher 50000 Bytes

 move.l #begr_txt,-(sp)       ;Bei der Installierung des Programms wird
 move.w #9,-(sp)              ;ein Text ausgegeben,  der gelesen werden
 trap #1                      ;kann, wenn eine Taste gedrueckt wird.
 addq.l #6,sp                 ;Textausgabe mit GEMDOS-PRINTLINE.

 move.l #300000,d1
lesen_abwarten:               ;Warteschleife  zum Lesen der  Aufforder-
                              ;ung,  zum   Weiterlesen  eine  Taste  zu
 subq.l #1,d1                 ;druecken.
 bne lesen_abwarten

 move.w #11,-(sp)             ;GEMDOS-CONSTAT: Ist eine Taste gedrueckt
 trap #1                      ;worden? Falls nicht, steht in d0 eine 0.
 addq.l #2,sp                 ;Dann wird nicht mehr gewartet.
 cmp.w #0,d0
 beq ende_lesen

 move.w #8,-(sp)              ;Im anderen Fall zwei Tasten abwarten mit
 trap #1                      ;GEMDOS-CONIN_WITHOUT_ECHO, eine ist  ja
 trap #1                      ;schon gedrueckt worden, die zweite been-
 addq.l #2,sp                 ;det das Lesen und es geht los.

ende_lesen:
 pea patch                    ;Die Routine patch wird  mit  der  XBIOS-
 move #38,-(sp)               ;Routine SUPEREXEC im Supervisormodus ab-
 trap #14                     ;gearbeitet.
 addq.l #6,sp

 move.l 4(sp),a0              ;GEMDOS-PTERMRES erhaelt das Programm,das
 move.l 12(a0),d2             ;verlassen wird, im Speicher resident.
 add.l  20(a0),d2
 add.l  28(a0),d2
 add.l  #$100,d2
 move.l d2,-(sp)
 move   #$31,-(sp)
 trap #1

patch:
 move.l $456,a0               ;Die Hardcopyroutine, die mit  Alternate/
 move.l #los_gehts,28(a0)     ;Help  aufgerufen wird, beginnt  bei  der
                              ;durch los_gehts bezeichneten Adresse.
return:
 rts
```

Bild 3. Das Assembler-Programm, das zu wesentlich besseren Hardcopies auf 24-Nadeldruckern führt

```
los_gehts:                    ;Dieser  Teil des Programms wird aufgeru-
 tst $4ee                     ;fen, wenn die  Alternate/Help-Taste  ge-
 bne return                   ;drueckt  wurde,  was  in  Speicherstelle
 move #$11,-(sp)              ;$4ee nochmal ueberprueft wird.
 trap #1                      ;Mit  GEMDOS-PRTOUT_STAT wird der Drucker
 add.l #2,sp                  ;kontrolliert.
 tst d0                       ;Ist  eine der Kontrollen  nicht  erfolg-
 beq return                   ;reich, wird das Programm beendet.

 move  #2,-(sp)               ;XBIOS-PHYSBASE ermittelt den Bildschirm-
 trap #14                     ;speicheranfang. Das Ende des Bildschirm-
 addq.l #2,sp                 ;speichers,  bei  dem  der Druck beginnt,
 move.l d0,a2                 ;liegt um 31999 Byte darueber.
 add.l #31999,a2              ;d1  ist  Zaehlvariable fuer die Zahl der
 move.l #79,d1                ;zu druckenden Zeilen.
 lea druckerreset,a1
 bsr string_aus               ;Der  Druckerreset wird mit  der  Routine
                              ;string_aus an den Drucker geschickt.
zeilenschleife:
 move.l a2,a3                 ;a3 adressiert den Bildschirmspeicher.
 sub.l d1,a3
 lea grafik,a1                ;Grafikmodus  im   Drucker   einschalten,
 bsr string_aus               ;Mehrfachdruck einer Zeile wird vorange-
                              ;stellt: 27,86,n. Die Zahl n ist 1,  wenn
                              ;sie beim Label grafik nicht anders fest-
                              ;gelegt wurde.  Das Ende der Grafikzeile,
                              ;die mehrfach gedruckt werden soll,  wird
                              ;spaeter mit multi_aus angezeigt.
 move.w #399,d2               ;Der Bildschirm hat 400 Grafikzeilen.  d2
                              ;ist Zaehlvariable dafuer.

byteschleife:                 ;Jedes  Bildschirmbyte wird in seine Bits
 clr.l d5                     ;zerlegt und in Relation zu seiner  Umge-
 clr.l d6                     ;bung  betrachtet.  Das  Bit muss dann in
 clr.l d7                     ;einen  Druckercode gewandelt werden,  so
                              ;dass es als 3x3 Nadelanschlaege ausgege-
                              ;ben werden kann. Ein  ganzes  Byte  wird
                              ;also mit 24x3 Nadeln dargestellt.
                              ;Die Register d5,d6 und d7 enthalten die-
                              ;se Information  waehrend der Berechnung.

;Das Byte im Bildschirmspeicher, das durch (a3) adressiert wird,  hat
;die in  der Skizze dargestellte Umgebung. Die Zahlen vor der Klammer
;sind der Index bezogen auf (a3) fuer die indizierte Adressierung.

;    ------------------  ----------------------------------------
;    ! Bit ! Bit ! Bit !  ! Bit ! Bit ! Bit ! Bit ! Bit ! Bit ! Bit !
;    !  0  !  7  !  6  !  !  5  !  4  !  3  !  2  !  1  !  0  !  7  !
;    !-81()!-80()!-80()!  !-80()!-80()!-80()!-80()!-80()!-80()!-79()!
;    !-----#############  ########################################-----!
;    ! Bit # Bit # Bit #  # Bit # Bit # Bit # Bit # Bit # Bit # Bit !
;    !  0  #  7  #  6  #  #  5  #  4  #  3  #  2  #  1  #  0  #  7  !
;    ! -1()#  () #  () #  #  () #  () #  () #  () #  () #  () # 1() !
;    !-----#############  ########################################-----!
;    ! Bit ! Bit ! Bit !  ! Bit ! Bit ! Bit ! Bit ! Bit ! Bit ! Bit !
;    !  0  !  7  !  6  !  !  5  !  4  !  3  !  2  !  1  !  0  !  7  !
;    ! 79()! 80()! 80()!  ! 80()! 80()! 80()! 80()! 80()! 80()! 81()!
;    ------------------  ----------------------------------------

;Bei der Berechnung wird immer nur Bit 7 des Bildschirmbytes mit  den
;8 Bit  seiner Umgebung verglichen, die 9 Nadelanschlaege fuer diesen
```

```
;Punkt ermittelt und in d5-d7 vermerkt. Um alle 8 Bit zu berechnen,
;muessen alle in der Skizze angegebenen Bits nach links verschoben
;werden und die Berechnung wird wiederholt. d5-d7 erfordern ebenfalls
;diese Verschiebung,,allerdings um 3 Bit.

 move.b -79(a3),al+2        ;Die Verschiebung kann nicht im Bild-
 move.b -80(a3),al+1        ;schirmspeicher erfolgen, weil er dadurch
 move.b -81(a3),al          ;veraendert wuerde. Daher wird der zu be-
 move.b 1(a3),am+2          ;arbeitende Bereich an die Stellen al, am
 move.b (a3),am+1           ;und ar im Speicher kopiert.
 move.b -1(a3),am
 move.b 81(a3),ar+2
 move.b 80(a3),ar+1
 move.b 79(a3),ar

 move.b #8,bitzaehler       ;Schleife, um 8 Bits und Umgebung zu te-
bits_testen:                ;sten. Falls das Bilschirmpixel geloescht
 btst #7,am+1               ;ist, zum Programmteil bit_geloescht ge-
 beq bit_geloescht          ;hen. Sonst werden die Bits 24-26 der
 ori.l #$07000000,d5        ;Register d5-d7 gesetzt und im folgenden,
 ori.l #$07000000,d6        ;je nach Umgebung, zum Teil wieder ge-
 ori.l #$07000000,d7        ;loescht.

 btst #0,al                 ;Ist die gesamte Umgebung des Pixels ge-
 bne s_loben                ;loescht, steht es also isoliert, wird
 btst #7,al+1               ;keine Korrektur vorgenommen und der Aus-
 bne s_loben                ;druck erfolgt mit 3x3 Nadeln.
 btst #6,al+1
 bne s_loben
 btst #0,am
 bne s_loben
 btst #6,am+1
 bne s_loben
 btst #0,ar
 bne s_loben
 btst #7,ar+1
 bne s_loben
 btst #6,ar+1
 bne s_loben
 btst #0,al
 bne s_loben
 btst #0,al
 bne s_loben
 jmp bit_weiter
s_loben:
 btst #0,am                 ;Die linke obere Ecke des Pixel grenzt an
 bne s_roben                ;drei andere. Wenn alle geloescht sind,
 btst #0,al                 ;wird der Anschlag in dieser Ecke wegge-
 bne s_roben                ;lassen, also Bit 24 von d7 geloescht.
 btst #7,al+1
 bne s_roben
 bclr #24,d7
s_roben:
 btst #7,al+1               ;Die rechte obere Ecke wird genauso ge-
 bne s_lunten               ;testet und ggf. Bit 26 von d7 geloescht.
 btst #6,al+1
 bne s_lunten
 btst #6,am+1
 bne s_lunten
 bclr #26,d7
s_lunten:
 btst #0,am                 ;Bit 24 von d5 legt fest, ob die linke
 bne s_runten               ;untere Ecke gedruckt wird.
 btst #0,ar
 bne s_runten
 btst #7,ar+1
 bne s_runten
 bclr #24,d5
s_runten:
 btst #7,ar+1               ;Rechts unten wird im Pixel kein Nadelan-
 bne bit_weiter             ;schlag erzeugt, wenn Bit 26 von d5 nicht
 btst #6,ar+1               ;gesetzt ist.
 bne bit_weiter
 btst #6,am+1
 bne bit_weiter
 bclr #26,d5
 jmp bit_weiter             ;Die Schleife wird bei bit_weiter fortge-
                            ;setzt.
bit_geloescht:
 btst #7,al+1               ;Das betrachtete Bildschirmpixel ist ge-
 beq c_loben                ;loescht. Falls die Nachbarpunkte links,
 btst #0,am                 ;rechts, oben und unten dagegen gesetzt
 beq c_loben                ;sind, gilt das Pixel als alleinstehender
 btst #6,am+1               ;weisser Fleck und wird nicht veraendert.
 beq c_loben
 btst #7,ar+1
 beq c_loben
 jmp bit_weiter
c_loben:
 btst #0,am                 ;Sind die Pixel links und oben gesetzt,
 beq c_roben                ;dann wird ein Nadelanschlag in diese Ek-
 btst #7,al+1               ;ke gesetzt. Bit 24 in Register d7 ist da
 beq c_roben                ;zustaendig.
 bset #24,d7
c_roben:
 btst #7,al+1               ;Ebenso wird rechts oben verfahren. Bit
 beq c_lunten               ;26 von d7 kann gesetzt werden.
 btst #6,am+1
 beq c_lunten
 bset #26,d7
c_lunten:
 btst #0,am                 ;Um die Ecke links unten im Pixel auszu-
 beq c_runten               ;fuellen falls die Nachbarpixel links und
 btst #7,ar+1               ;rechts gesetzt sind, muss Bit 24 von d5
 beq c_runten               ;gesetzt werden.
 bset #24,d5
c_runten:
 btst #7,ar+1               ;Fuer rechts unten gilt es entsprechend,
 beq bit_weiter             ;Bit 26 von Register d5 zu setzen.
 btst #6,am+1
 beq bit_weiter
 bset #26,d5
bit_weiter:
 ror.l #3,d5                ;Sowohl die Speicher fuer die Nadelan-
 ror.l #3,d6                ;schlaege d5-d7 als auch die Zwischen-
 ror.l #3,d7                ;speicher des Bildschirmbytes und seiner
 move.l al,d0               ;Umgebung al, am und ar rotieren.
 rol.l #1,d0
 move.l d0,al
 move.l am,d0
 rol.l #1,d0
 move.l d0,am
```

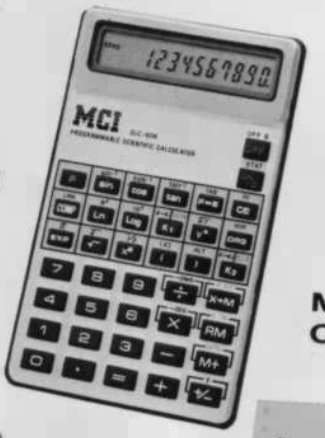
```
 move.l ar,d0
 rol.l #1,d0
 move.l d0,ar
 subq.b #1,bitzaehler       ;Falls noch nicht alle 8 Bits des Bild-
 bne bits_testen            ;schirmbytes bearbeitet worden sind, wird
                            ;die Schleife bits_testen durchlaufen. In
                            ;d5-d7 stehen anschliessend die 9 Byte,
                            ;die der Drucker zur Darstellung der acht
                            ;Pixel benoetigt. Allerdings werden die
                            ;Bytes vom Drucker umgekehrt, das heisst
 clr.l d4                   ;von Bit 7-0, interpretiert. Jedes Byte
 move.b d5,d4               ;muss also gespiegelt werden. Die Routine
 jsr spiegeln_aus           ;spiegeln_aus uebernimmt das und sendet
 ror.l #8,d5                ;es an den Drucker.
 move.b d5,d4
 jsr spiegeln_aus
 ror.l #8,d5
 move.b d5,d4
 jsr spiegeln_aus
 move.b d6,d4
 jsr spiegeln_aus
 ror.l #8,d6
 move.b d6,d4
 jsr spiegeln_aus
 ror.l #8,d6
 move.b d6,d4
 jsr spiegeln_aus
 move.b d7,d4
 jsr spiegeln_aus
 ror.l #8,d7
 move.b d7,d4
 jsr spiegeln_aus
 ror.l #8,d7
 move.b d7,d4
 jsr spiegeln_aus
 sub.l #80,a3               ;Das naechste Byte wird aus der vorherge-
 dbra d2,byteschleife       ;henden Bildschirmzeile geholt. a3=a3-80.
                            ;Wenn d2 400 Bilschirmbyte gezaehlt hat,
                            ;wird mit einer neuen Druckerzeile begon-
                            ;nen, sonst kommt das naechste Byte.
 lea multi_aus,a1           ;Ende der Druckwiederholung. Die Druck-
  bsr string_aus            ;zeile wird sooft gedruckt, wie an der
                            ;Stelle mit dem Label grafik mit der Fol-
                            ;ge 27,86,n festgelegt wurde. Der Wert
                            ;kann da im Listing geaendert werden.
 lea linefd,a1              ;Zeilenvorschub.
 bsr string_aus
 tst $4ee                   ;Hardcopy beenden?
 dbne d1,zeilenschleife     ;Falls nicht und auch die 80 Druckzeilen
                            ;in d1 noch nicht heruntergezaehlt worden
 move #-1,$4ee              ;sind, naechste Druckzeile.
 lea normal,a1              ;In $4ee soll eine -1 stehen, um die Al-
 bsr string_aus             ;ternate/Help-Taste zu erkennen.
                            ;Drucker wieder normal einstellen und zu-
 rts                        ;rueck.

                            ;       Unterprogramm string_aus
string_aus:                 ;In a1 steht die Adresse des Textes.
 move.b (a1)+,d3            ;string_aus liest zuerst die Zahl der
 ext.w d3                   ;auszugebenden Zeichen in d3. Sooft wird
 subq #1,d3                 ;dann an den Drucker gesendet.
string_aus_loop:
 move.b (a1)+,d4
 bsr zeichen_aus
 dbra d3,string_aus_loop
 rts
 zeichen_aus:               ;       Unterprogramm zeichen_aus
  move d4,-(sp)             ;zeichen_aus benutzt GEMDOS-PRTOUT zur
  move #5,-(sp)             ;Ausgabe des in d4 vorliegenden Codes an
  trap #1                   ;den Drucker.
  addq.l #4,sp
  rts

                            ;       Unterprogramm spiegeln_aus
 spiegeln_aus:              ;Die Routine spiegeln_aus vertauscht fuer
  move.b d4,d0              ;d4 Bit 7 und Bit 0, Bit 6 und Bit 1 und
  move.b #8,bitzaehler      ;so weiter. Das gespiegelte Byte wird an-
 spiegel_loop:              ;schliessend an den Drucker geschickt.
  roxr.b #1,d0
  roxl.b #1,d4
  subq.b #1,bitzaehler
  bne spiegel_loop
  jsr zeichen_aus
  rts


  data
 multi_aus: dc.b 4,13,27,86,0                ;Endemarkierung Mehrfach-
                                             ;druck einer Zeile.
 druckerreset:
            dc.b 2,28,64                     ;Ruecksetzen des Druckers.
 grafik:    dc.b 8,27,86, 1 ,27,"*",39,176,4 ;Die ersten 3 Codes fueh-
                                             ;ren zum 1-fachen Druck.
                                             ;Hier kann der Mehrfach-
                                             ;druck eingestellt werden.
 linefd:    dc.b 4,27,"3",24,10              ;Zeilenvorschub.
 normal:    dc.b 7,13,10,27,"A",12,27,"2"    ;Drucker wieder normal.
  align
 al:        dc.l 0                           ;a1, am und ar sind Zwi-
 am:        dc.l 0                           ;schenspeicher der Bild-
 ar:        dc.l 0                           ;schirminformation.
 bitzaehler:dc.l 0                           ;Zaehlvariable.
 begr_txt:
  dc.b 27,69,13,10,10,10,32,27,112
  dc.b "                                ZUM LESEN TASTE         "
  dc.b "                               ",27,113,10,10,10,13
  dc.b "                  Hardcopy fr den NEC P6/P7 mit weichen Konturen"
  dc.b 10,10,13
  dc.b "                                                von",10,13
  dc.b "                                      Erich Ludwig, Paderborn",10,13
  dc.b "                                           Januar 1988",13,10,10
  dc.b "     Dieses Programm druckt den Bildschirm ab, wenn die"
  dc.b " Alternate/Help-Taste",10,13
  dc.b "     gedrueckt wird. Diagonale Linien werden weniger stufig"
  dc.b " dargestellt.",10,10,13
  dc.b "     Mit Alternate/Help kann der Druckvorgang abgebrochen"
  dc.b " werden.",10,10,13
  dc.b "                                                   Taste",0

    end
```

KOMPLETTPREISE...SYSTEMPAKETE... 1 JAHR GARANTIE ... KOMPLETTPREISE...SYSTEMPAKETE

System Pakete für kluge Rechner

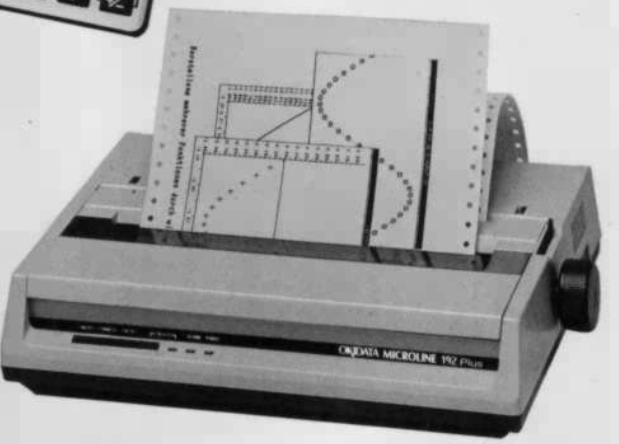
ab 1349,-

+



(+)

MCI Printer Plus a. A.
OKI ML192 Elite a. A.



System Paket 11

10 MHz

1349,-

- **MCI XT16SLC,** 640 K, 1 x 360 K, Clock, ser. par., 12" Monitor, Tastatur
- **MCI** Programmierbarer Taschenrechner

optional MCI 120 Zeichen Printer oder OKI ML 192 Elite

System Paket 21

10 MHz

1899,-

- **MCI XT16SLC,** 640 K, 1 x 360 K, Clock, ser. par., 12" Monitor, Tastatur
- **MCI** 20MB Festplatte
- **MCI** Programmierbarer Taschenrechner

optional MCI 120 Zeichen Printer oder OKI ML 192 Elite

System Paket 31

12 MHz

2399,-

- **MCI AT4SLC,** 640 K, 1 x 1, 2 MB, Clock, ser. par., 12" Monitor, Tastatur
- **MCI** Programmierbarer Taschenrechner

optional MCI 120 Zeichen Printer oder OKI ML 192 Elite

System Paket 41

12 MHz

3199,-

- **MCI AT4SLC,** 640 K, 1 x 1, 2 MB, Clock ser. par., 12" Monitor, Tastatur
- **MCI** 20MB Festplatte
- **MCI** Programmierbarer Taschenrechner

opt. MCI 120 Zeichen Printer oder OKI ML 192 Elite

PREIS...QUALITÄT...1 JAHR GARANTIE

KOMPATIBEL...24-STUNDEN-TEST...LEISTUNG...PREIS

MCI XT 16 SLC

alles drin!

Grundausstattung ohne Monitor

ab **899,–**

beinhaltet:

- voll IBM®XT kompatibel
- 8088 CPU + 8087 Sockel
- 8 XT Slots
- 256 KB freier Speicher
- 1 x 360 KB Floppy-Drive
- Color- oder Monochr. Grafikkarte (Hercules II komp. 720 x 348 P.)
- Deutsche Normtastatur MK 5111
- 150 W Schaltnetzteil
- Parallele Drucker-Schnittstelle

Erweiterungen für XT 16 SLC-Serie

2. Laufwerk 360 KB	249,–
Speichererweiterung auf 640 KByte	a. A.
Clock/Seriell-Karte	79,–
I/0 Plus II Karte	149,–
20 MB Festplatte mit XT-Controller	+ 589,–
30 MB Festplatte m. RLL XT-Contr.	+ 679,–
EGA-Set statt monochr. Karte	+ 1149,–
Opt. Maus mit Tablet	119,–
MS-DOS 3.3 deutsch + GW-Basic	+ 199,–
Professional Multifunktions-Tastatur MK 6000	+ 100,–
9" TTL Monitor grün	+ 150,–
12" Monitor grün od. bern.	+ 229,–
14" TTL Monitor grün, bern. od. weiß	+ 279,–

MCI AT 4 SLC

alles drin!

Grundausstattung ohne Monitor

ab **1899,–**

beinhaltet:

- voll IBM® AT kompatibel
- 80286 CPU + 80287 Sockel
- 6 AT + 2 XT Slots
- 8 und **12 MHz** umschaltbar
- 512 KB freier Speicher
- 1 x 1,2 MB/360 KB Laufwerk
- Color- oder Monochr. Grafikkarte (Hercules II komp. 720 x 348 P.)
- Parallele Drucker-Schnittstelle
- Batteriegep. Echtzeituhr/Kalender
- Kapazitive deutsche Normtastatur

Erweiterungen für AT 4 SLC-Serie

2. Laufwerk 360 KB	299,–
20 MB Festplatte mit AT-Controller	889,–
30 MB Festplatte m. RLL AT-Contr.	+ 999,–
Seriell-Karte	79,–
I/0 Plus II Karte	149,–
EGA-Set statt monochr. Karte	+ 1149,–
MS-DOS 3.3 deutsch + GW-Basic	+ 199,–
Professional Multifunktions-Tastatur MK 6000	+ 100,–
9" TTL Monitor grün	+ 150,–
12" Monitor grün od. bern.	+ 229,–
14" TTL Monitor grün, bern./weiß	+ 279,–

PRINTER

MCI *Personal Computer Graphics Printer Plus*

- **voll kompatibel zum** *IBM Personal Computer Graphics Printer*
- **120 Zeichen/sec.**

a. A.

OKI MICROLINE ML 192 Elite

- 9 Nadel Matrixdrucker
- Druckgeschwindigkeit 200 Z./sec.
- 40 Zeichen/sec. NLQ
- Druckpuffer 8 KB
- IBM Kompatibel

a. A.

EGA

Hochauflösender EGA-Monitor

EGA-Karte 249,–

- Auflösung
  320 x 200 (CGA Mode)
  640 x 350 (EGA Mode)

999,–

TELEFON-HOTLINE-PREISE
MCI Telefonansage (20 sec.)

	(0 22 02)
Festplatten & Controller	10 81 40
Grafikkarten	10 81 41
Monitore	10 81 42
Schnittstellenkarten	10 81 43
Drucker	10 81 44
Aktuelle Neuigkeiten	10 81 45

MCI

5060 Bergisch Gladbach 2
Bensberger Straße 252
Tel.-Nr.: 02202/1080
Fax: 02202/31009 · Telex: 8873518

12 Monate Garantie auf alle Geräte. Nach der Pang Vo. v. 14. 3. 85 sind wir bei Angeboten gegenüber dem Endverbraucher zur Angabe der **Preise incl. MwSt.** verpflichtet. Für Druckfehler wird nicht gehaftet. **Preise gültig ab 1. 7. 88.** Lieferzeit und Lieferbedingungen auf Anfrage. Änderungen, die technischen Verbesserungen dienen, vorbehalten. Zwischenverkauf vorbehalten.
MCI MICRO COMPUTER INSTRUMENTS GMBH eingetragen AG Bergisch Gladbach · HRB 2575. Herstellung und Vertrieb von Mikrocomputern. 5060 Bergisch Gladbach 2 · Bensberger Straße 252

Christian Brunner

Der Maus-Interrupt 33H

Programme mit Maus-Routinen besser bedienbar gemacht

Bei Standard-Softwarepaketen gibt es mit Billig-Mäusen nur selten Probleme. Anwender, die ein selbst geschriebenes Programm mit der Maus bedienen wollen, stehen dagegen im Regen, da fast kein Hersteller dokumentiert, wie man mit den vom Treiber bereitgestellten Funktionen umgehen kann.

In der Dokumentation von Billig-Mäusen findet man oft kein Wort darüber, wie man die Funktionen des Maustreibers nutzen kann. Beim Starten des Treibers – er heißt meist MOUSE.COM – verbleibt dieser resident im Speicher und stellt eine BIOS-Erweiterung dar, die über den Interrupt 33H erreichbar ist. Bei seriellem Anschluß der Maus – was bei preisgünstigen kompatiblen Mäusen die Regel ist – wird außerdem noch der Interruptvektor, der zuvor auf die V.24-Zeichenemfangsroutine im BIOS gezeigt hat, auf die Adresse einer entsprechenden Routine innerhalb des Treibers „verbogen". Damit kann der Rechner jedes Zeichen interpretieren, das die Maus bei der Änderung ihres Zustandes sendet.

Die serielle Schnittstelle COM1 wird von der Interrupt-Service-Routine des Interrupts 0CH bedient. Für COM2 ist der Interrupt 0BH zuständig. Ferner wird noch der Video-Interrupt 10H manipuliert, da der Treiber in der Lage ist, einen Cursor über den Bildschirm zu bewegen.

Das Übertragungsformat der seriellen Maus

Wird die Maus bewegt oder eine Taste gedrückt bzw. losgelassen, überträgt die Maus einen Block bestehend aus fünf Zeichen. Ein Zeichen wird wie folgt übertragen: 1 Startbit, 8 Datenbits, 1 Stoppbit, keine Parität, die Übertragungsrate beträgt 1200 Bit/s. Das erste Zeichen innerhalb des Blocks ist ein Synchronisations-Zeichen, das bereits den Zustand der Maustasten wiederspiegelt (*Bild 1*). Darauf folgen vier Zeichen, die als relative Positionsangaben aufgefaßt werden können. Sie enthalten die Positionsänderung der Maus in X- und Y-Richtung seit der letzten Übertragung in der Reihenfolge DeltaX1, DeltaY1, DeltaX2, DeltaY2. Die Angaben werden dabei in hundertstel Zoll gemacht, d. h. ein Schritt entspricht 0.254 mm. Ein Treiber-Programm, das direkt diesen Interrupt bedient, muß sowohl die X1/Y1- als auch die X2/Y2-Werte als voll gültig behandeln. Erst wenn alle fünf Zeichen empfangen sind, ist der Datenblock als beendet anzusehen. Alle eventuell folgenden Zeichen sind zu ignorieren, bis das Synchronisationszeichen erneut eintrifft und der Zyklus von vorne beginnt. Zu beachten ist nur noch, daß die „Delta-Zeichen" vorzeichenbehaftet sind, wobei ein positiver X-Wert eine Bewegung nach rechts, ein positiver Y-Wert eine Bewegung nach oben repräsentiert. Der Wertebereich geht demnach von −128 bis +127.

Aufruf der Funktionen des Maustreibers

Die einzelnen Routinen des mitgelieferten Treibers sind genauso zu erreichen, wie es bei den übrigen BIOS-Funktionen von IBM-kompatiblen PCs üblich ist. In AX übergibt man die Funktionsnummer, etwaige Parameter werden in den Registern BX, CX und DX erwartet. Entsprechend findet man Ergebniswerte in den gleichen Registern. Um mögliche Abstürze zu vermeiden, sollte jedes Anwenderprogramm schon beim Start überprüfen, ob der Maustreiber überhaupt geladen ist. Dies kann man als gegeben ansehen, wenn weder der Eintrag für den Interrupt 33H in der System-Interrupttabelle (Adresse 0000:n*4, n = Interruptnummer, hier 33H) aus vier Nullen besteht, noch die durch diesen Eintrag adressierte Speicherzelle einen Interrupt-Return-Befehl (IRET, Op-Code 0CFH) enthält. Weiterhin ist es sinnvoll, festzustellen, ob die Maus angeschlossen ist, dazu ist die Funktion 0 wie unten erklärt auszuführen. Wie man eine Test-Routine schreibt, die testet ob der Maustreiber bereits geladen ist, sowie die Durchführung eines Funktionsaufrufes in Basic ist *Bild 2* zu entnehmen. Der Aufruf einer Funktion geschieht wie folgt:

CALL MOUSE (P1%,P2%,P3%,P4%).

Die vier Parameter müssen auf jeden Fall Integervariablen sein. Am sichersten ist es daher, ganz einfach Variablennamen mit angehängtem Prozentzeichen zu verwenden. Ergebnisse erhält man wieder in P1%...P4% zurück. Im Turbo-Pascal-Beispiel heißt die Test-Funktion „MouseDriverLoaded", die Aufruf-Prozedur „Mouse (m1,m2, m3,m4: Integer)".

Die Funktionen des Treibers

Ein „Minimaltreiber", wie er bei den meisten kompatiblen Mäusen mitgeliefert wird, kennt im allgemeinen nur die ursprünglich vorgesehenen ersten 16 Funktionen; neuere Treiber von Microsoft, Logitech und IBM kennen oft noch zusätzliche Erweiterungen, die jedoch zum Teil herstellerspezifisch sind. Zwar mögen die zugefügten Funktionen von Typ zu Typ ähnlich sein, doch müssen die Funktionsnummern und -parameter keineswegs übereinstimmen. Man findet z. B. Routinen zum Sichern und Zurückholen der aktuellen Treibereinstellungen oder zur Manipulation der EGA-Register. Bei Mäusen, deren Treiber erweiterte Funktionen aufweisen, wird aber in der Regel eine entsprechende Dokumentation beigelegt. Oft sind zumindest erweiterte Möglichkeiten bei der Darstellung des Grafik-Cursors implementiert (Funktion 18) oder auch eine Routine zum Schützen eines Bildschirmfensters vor dem Cursor (Funktion 16); deshalb werden diese Funktionen weiter unten näher beschrieben. Es folgt eine Aufzählung der Standardfunktionen. Unter Eingabe stehen die Aufrufparameter, unter Rückgabewerte die zurückgelieferten Werte.

Funktion 0: Rücksetzen
Eingabe: AX = 0.
Rückgabewerte: AX = -1 = FFFFH, falls Maus und serielle Karte angeschlossen, sonst 0,
BX = Maustyp: 2 = Microsoft Mouse, 3 = Mouse Systems Mouse.
Der serielle Port der Maus wird auf das richtige Übertragungsformat programmiert und die durch andere Funktionen veränder-

baren Einstellungen des Treibers werden wie folgt initialisiert:
- horizontale Empfindlichkeit (Funktion 15) auf 8 einstellen
- vertikale Empfindlichkeit (Funktion 15) auf 16 einstellen
- der Grafik-Cursor wird zu einem Pfeil und dessen Bezugspunkt (Funktion 9) wird auf (-1, -1) gesetzt
- der Text-Cursor wird zu einem Block
- Grenzen für die Cursorposition (Funktion 7 und 8) einstellen: Xmin = 0, Xmax = 639, Ymin = 0, Ymax = 199
- Zeiger auf eine Benutzerroutine (Funktion 12) löschen
- Light-Pen Emulation (Funktionen 13 und 14) erlauben

Die Funktion 0 ist nicht in der Lage, die Hercules-Monochrom-Karte automatisch zu erkennen. Um die Funktion der Darstellung des Grafik-Cursors zu gewährleisten, muß sich die Karte vor dem Aufruf der Funktion 0 im Grafikmodus befinden und der Programmierer muß in die Adresse des Bildschirmmodus-Bytes ($40:$49) den Wert 6 (für Hercules-Seite 0, Segment bei $B000) bzw. 5 (für Hercules-Seite 1, Segment bei $B800) schreiben.

Funktion 1: Cursordarstellung freigeben
Eingabe: AX = 1.
Rückgabewerte: keine.
Je nachdem, in welcher Bildschirmbetriebsart der Rechner sich gerade befindet, wird ein Text- oder Grafik-Cursor eingeblendet. Verschiebt man die Maus, wird entsprechend ihrer Bewegung der Cursor mitgeführt. Dies ist sehr praktisch, denn somit erfolgt jede Bewegung unabhängig von der jeweiligen Anwendung. Ausgenommen sind nur die Augenblicke, in denen Interrupts gesperrt sind. Diese Zeitdauer ist jedoch normalerweise so kurz, daß man keine Unregelmäßigkeiten in der Cursorbewegung bemerkt.

Funktion 2: Cursordarstellung sperren
Eingabe: AX = 2.
Rückgabewerte: keine.
Hiermit wird der Grafik- oder Text-Cursor ausgeblendet, der Treiber jedoch arbeitet intern nach wie vor mit Cursorkoordinaten weiter. Immer, wenn man die unter dem Cursor liegende Grafik verändert, muß man die Darstellung des Cursors sperren, sonst kopiert der Maustreiber beim Bewegen des Cursors den alten Bildschirmausschnitt ein.

Funktion 3: Maus-Cursor-Position und Tastenzustand abfragen
Eingabe: AX = 3.
Rückgabewerte: BX = Tastenzustand,

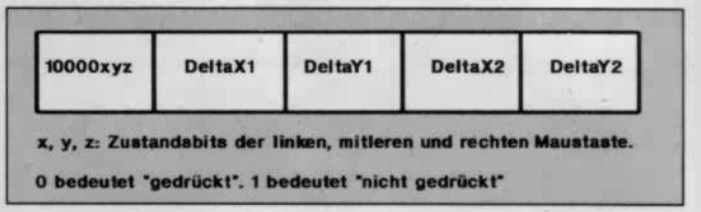


Bild 1. Der 5-Byte-Datenblock der Maus

```
10 DEF SEG = 0 : INT33V = &H33 * 4 : INTRET = &HCF
20 MOUSESEG = PEEK (INT33V+2) + 256 * PEEK (INT33V+3):REM Maustreiber-Offset
30 MOUSE    = PEEK (INT33V)   + 256 * PEEK (INT33V+1) + 2:REM Einsprungsadresse
40 DEF SEG = MOUSESEG
50 IF MOUSESEG<>0 AND MOUSE<>2 AND PEEK (MOUSE-2)<>INTRET THEN 80
60 PRINT "Der Maustreiber ist nicht installiert!"
70 STOP
80 REM Maustreiber ordnungsgemäß vorhanden.
```

Bild 2. Das kleine Basic-Programm prüft ob der Maus-Treiber bereits geladen ist

CX = logische X-Position des Maus-Cursors,
DX = logische Y-Position des Maus-Cursors.
In BX entsprechen die Bits 0, 1 und 2 der linken, rechten und mittleren Maustaste. Dabei bedeutet 1 „Taste gedrückt" und 0 „Taste nicht gedrückt". CX und DX liegen immer innerhalb des Wertebereiches, der durch den jeweiligen Bildschirmmodus bestimmt wird. Manchmal muß man, um die wirklichen physikalischen Pixelkoordinaten (bzw. im Textmodus die Textzeichenkoordinaten) zu erhalten, noch den CX- und/oder DX-Wert durch einen Divisor teilen, wie in *Tabelle 1* gezeigt.

Funktion 4: Maus-Cursor-Position setzen
Eingabe: AX = 4, CX = gewünschte logische X-Position des Maus-Cursors,
DX = gewünschte logische Y-Position des Maus-Cursors.
Rückgabewerte: keine.
Mit dieser Funktion wird der Cursor neu positioniert, CX und DX müssen innerhalb des gültigen logischen Bereiches liegen (siehe Tabelle 1). Bei eingeschaltetem Cursor wird die Positionsänderung am Bildschirm nicht angezeigt. Ein einfaches Mittel, um dies zu erreichen, ist ihn aus- und sofort wieder einzuschalten.

Funktion 5: Informationen über das Drükken einer bestimmten Taste holen
Eingabe: AX = 5, BX = 0 ,1 oder 2 entsprechend linker, rechter oder mittlerer Taste.
Rückgabewerte: AX = Tastenzustand zur Zeit des Aufrufes (Codierung wie im BX-Register bei Fuktion 3)
BX = Anzahl der Betätigungen der selektierten Taste seit der letzten Abfrage durch Funktion 5,
CX = logische X-Position des Maus-Cursors bei letztem Tastendruck,
DX = logische Y-Position des Maus-Cursors bei letztem Tastendruck.
Der Maustreiber zählt die Betätigungen der drei (oder zwei) Maustasten mit. Diese Zahl kann man für jede Taste durch Funktion 5 abrufen, der Wertebereich liegt zwischen 0 und 32767. Nach dem Aufruf ist der selektierte Betätigungszähler wieder auf Null gesetzt. Hierbei kann man aus CX und DX die X/Y-Koordinaten des Cursors bei der letzten Betätigung der entsprechenden Taste erfahren.

Funktion 6: Informationen über das Loslassen einer bestimmten Taste holen
Eingabe: AX = 6, BX = 0, 1 oder 2 entsprechend linker, rechter oder mittlerer Taste.
Rückgabewerte: siehe Funktion 5.
Einziger Unterschied: diese Funktion bezieht sich nicht auf das Niederdrücken, sondern auf das Loslassen einer Taste.

Funktion 7: Minimale und maximale X-Position des Cursors setzen
Eingabe: AX = 7, CX = minimale logische X-Koordinate des Cursors,
DX = maximale logische X-Koordinate des Cursors.
Rückgabewerte: keine.
Die in der Tabelle 1 dargestellten logischen Grenzen des Cursors können mittels der Funktionen 7 und 8 verändert werden. Somit kann man sehr bequem ein Bildschirmfenster definieren, in dem sich der Cursor bewegen darf. Befindet er sich zur Zeit des Aufrufes dieser oder der nächsten Funktion außerhalb des neuen Fensters, bewegt der Maustreiber ihn an den Rand des Fensters.

Funktion 8: Minimale und maximale Y-Position des Cursors setzen
Eingabe: AX = 8,
CX = minimale logische Y-Koordinate des Cursors,
DX = maximale logische Y-Koordinate des Cursors.

Das unter Funktion 7 gesagte gilt hier entsprechend für die Y-Koordinaten.
Rückgabewerte: keine.

Tabelle 1: Arbeitshilfe zur Berechnung der Maus-Cursor-Position

Modus	physikalisches				logisches				Divisor X,	
	X_{min}	X_{max}	Y_{min}	Y_{max}	X_{min}	X_{max}	Y_{min}	Y_{max}	Divisor Y	
Text: 80×25	0	79	0	24	0	632	0	192	8	8
Text: 40×25	0	39	0	24	0	624	0	192	16	8
CGA: 640×200, 2 Farben	0	639	0	199	0	639	0	199	1	1
CGA: 320×200, 4 Farben	0	319	0	199	0	638	0	199	2	1
PCjr: 320×200, 16 Farben	0	319	0	199	0	1276	0	199	4	1
EGA-D: 320×200, 16 Farben	0	319	0	199	0	319	0	199	1	1
EGA-E: 640×200,16 Farben	0	639	0	199	0	639	0	199	1	1
EGA-F:640×350,monochrom	0	639	0	349	0	639	0	349	1	1
EGA-10: 640×350,4 Farben	0	639	0	349	0	639	0	349	1	1
Hercules:720×348,mono.	0	719	0	347	0	639	0	349	1	1

Funktion 9: Aussehen des Grafik-Cursors einstellen
Eingabe: AX = 9,
BX = X-Koordinate des Cursor-Bezugspunktes (-16 ... +16),
CX = Y-Koordinate des Cursor-Bezugspunktes (-16 ... +16),
DX = Zeiger auf die Cursor-Bitmap.
Rückgabewerte: keine.
Hiermit kann man die Form und Farbe des Grafik-Cursors, der vom Treiber automatisch mitgeführt wird, definieren. Standard ist der bekannte Pfeil von rechts unten nach links oben. BX und CX enthalten die Koordinaten desjenigen Pixels (picture element: Bildpunkt), der für die Positionsangabe des Cursors entscheidend ist. Bei der Pfeilform ist dies der Punkt links oberhalb der Pfeilspitze. Da BX und CX als Offset von dem links oben liegenden Punkt der Bitmap des Cursors aus gerechnet werden, sind in diesem Fall CX = -1 und DX = -1 gesetzt. Die Bitmap selbst besteht aus zwei Teilen: der sogenannten Und- und der Xor-Maske, die sich hintereinander an der Stelle [DX] im Speicher befinden müssen. Jede Maske besteht aus 16 Wörtern zu je 16 Bit, wobei ein Pixel wie folgt codiert ist: Im hochauflösenden CGA-Modus, im Hercules-Modus und in allen EGA-Modi entspricht ein Pixel einem Bit. Die Farbe läßt sich durch Aktivierung entsprechender Farbebenen nur für den ganzen Cursor ändern. Mehrere Farben innerhalb des Cursors sind einzig im 320 × 200 CGA-Modus einstellbar, dann entsprechen einem Pixel eben zwei Bits, mit denen sich vier Farben darstellen lassen. Die Und-Maske hat diesen Namen, weil sie mit den Pixels an der Cursorposition logisch verundet wird. Nachdem die Und-Maske auf diese Weise in den Bildschirmspeicher gebracht worden ist, wird an derselben Stelle noch die Xor-Maske mit den Punkten Exklusiv-Oder-verknüpft. Das entspricht der Invertierung jedes Bildschirmpunktes, dessen zuständiges Bit in der Xor-Bitmap gesetzt ist. Das Beispiel in *Bild 3* zeigt die Bitmap eines hellen Fadenkreuzes, das auf hellem Untergrund einen dunklen Rand bekommt. Ein Beispiel zur Programmierung in Basic gibt *Bild 4.* Das gleiche in Turbo-Pascal ist in *Bild 5* dargestellt. *Tabelle 2* erleichtert den Entwurf von Bitmaps.

Tabelle 2: Tips zur Programmierung von Bitmaps

gewünschter Bildpunkt	zugehöriges Bit in der AND-Maske	zugehöriges Bit in der XOR-Maske
unverändert	setzen	löschen
invertiert	setzen	setzen
dunkel	löschen	löschen
hell	löschen	setzen

Funktion 10: Text-Cursor einstellen
Eingabe: AX = 10, BX = 0 für Software-Cursor,
CX = Und-Maske, DX = Xor-Maske,
oder BX = 1 für Hardware-Cursor,
CX = erste dargestellte Cursorzeile,
DX = letzte dargestellte Cursorzeile.
Rückgabewerte: keine.
Ebenso wie beim Grafik-Cursor gibt es auch beim Text-Software-Cursor (BX = 0) eine Und- und eine Xor-Maske. Nur beziehen sich deren Bits nicht auf Pixel, sondern auf diejenigen 16 Bits, die im Textmodus ein Zeichen einnimmt:
Bits 0...7: Zeichen-Code
Bits 8, 9, 10: Zeichen-Farbe
Bit 11: Helligkeit (0 = dunkel, 1 = hell)
Bits 12, 13, 14: Hintergrund-Farbe
Bit 15: Blinken (0 = ein, 1 = aus)
Wenn man z. B. ein blinkendes Gesicht über den Bildschirm gleiten lassen will, setzt man CX auf 0, und DX auf $8001.
Beim Text-Hardware-Cursor (BX = 1) bestimmen CX und DX die Nummer der ersten und die der letzten dargestellten Cursorzeile. Für einen Blockcursor ist CX = 0 und DX = 11, für eine Unterstreichung ist CX = 10 und DX = 11 usw.

Funktion 11: Maus-Zählerstand abfragen
Eingabe: AX = 11.
Rückgabewerte: CX = X-Zählerstand,
DX = Y-Zählerstand.
Bei jedem Aufruf der Funktion 11 erhält man die Differenz der Koordinaten der aktuellen Mausposition und der Position beim letzten Aufruf, d. h. die Verschiebung der Maus in X- und Y- Richtung gegenüber der letzten Abfrage von Funktion 11. Der Wertebereich reicht von −32768...32767, positive Werte bedeuten Bewegung nach unten oder rechts.

Funktion 12: Zusätzliche Benutzer-Interruptroutine installieren
Eingabe: AX = 12, CX = Ereignismaske,
ES:DX = Adresse der benutzerspezifischen Zusatzroutine.
Rückgabewerte: keine.
Bei jeder Zustandsänderung (Position, Tastenstatus) der Maus wird die Adresse ES:DX angesprungen, wenn entsprechende Bits in der Ereignismaske CX gesetzt sind:

Bit 0: Änderung der Maus-Cursor-Position
Bit 1: linke Taste wurde gedrückt
Bit 2: linke Taste wurde losgelassen
Bit 3: rechte Taste wurde gedrückt
Bit 4: rechte Taste wurde losgelassen
Bit 5: mittlere Taste wurde gedrückt
Bit 6: mittlere Taste wurde losgelassen
Bits 7...15 sind bedeutungslos.

Setzt man also nur die Bits 1 und 3 in CX, wird das Unterprogramm bei [EX:DX] immer angesprungen, wenn die linke oder rechte Taste gedrückt wird.
Dabei sind die Prozessorregister wie folgt belegt:

AX: Byte, das den Grund des Ansprunges angibt, Codierung wie bei der Ereignismaske
BX: Tastenzustand (wie BX bei Funktion 3)
CX: logische X-Koordinate des Maus-Cursors
DX: logische Y-Koordinate des Maus-Cursors
DI: X-Wert des Mauszählerstandes (wie CX bei Funktion 11)
SI: Y-Wert des Mauszählerstandes (wie DX bei Funktion 11)

Zur Rückkehr aus der Benutzerroutine muß ein Far-Return-Befehl verwendet werden, da sie vom Treiber durch einen Far-Call angesprungen wird.

Funktion 13: Light-Pen-Emulation einschalten
Eingabe: AX = 13
Rückgabewerte: keine.
Um die Maus mit Programmen, die einen Lichtgriffel benötigen, verwenden zu können, muß die Emulation eingeschaltet sein (default nach Funktion 0). Als Beispiel sei die Funktion PEN in Basic genannt. Das Niederdrücken des Griffels auf den Schirm

```
"AND"-Maske (Fadenkreuz-Umriß):
 0: 1111111000111111 = $FE3F
 1: 1111111000111111 = $FE3F
 2: 1111111000111111 = $FE3F
 3: 1111111000111111 = $FE3F
 4: 1111111000111111 = $FE3F
 5: 1111111000111111 = $FE3F
 6: 1000000000000000 = $8000
 7: 1000000000000000 = $8000
 8: 1000000000000000 = $8000
 9: 1111111000111111 = $FE3F
10: 1111111000111111 = $FE3F
11: 1111111000111111 = $FE3F
12: 1111111000111111 = $FE3F
13: 1111111000111111 = $FE3F
14: 1111111000111111 = $FE3F
15: 1111111111111111 = $FFFF
```

```
"XOR"-Maske (Fadenkreuz-Inneres):
 0: 0000000000000000 = $0000
 1: 0000000010000000 = $0080
 2: 0000000010000000 = $0080
 3: 0000000010000000 = $0080
 4: 0000000010000000 = $0080
 5: 0000000010000000 = $0080
 6: 0000000010000000 = $0080
 7: 0011111111111110 = $3FFE
 8: 0000000010000000 = $0080
 9: 0000000010000000 = $0080
10: 0000000010000000 = $0080
11: 0000000010000000 = $0080
12: 0000000010000000 = $0080
13: 0000000010000000 = $0080
14: 0000000000000000 = $0000
15: 0000000000000000 = $0000
```

Bild 3. Bitmap eines Fadenkreuzes

```
99  REM AND-Maske:
100 GRCRSR( 0) = &HFE3F
110 GRCRSR( 1) = &HFE3F
120 GRCRSR( 2) = &HFE3F
130 GRCRSR( 3) = &HFE3F
140 GRCRSR( 4) = &HFE3F
150 GRCRSR( 5) = &HFE3F
160 GRCRSR( 6) = &H8000
170 GRCRSR( 7) = &H8000
180 GRCRSR( 8) = &H8000
190 GRCRSR( 9) = &HFE3F
200 GRCRSR(10) = &HFE3F
210 GRCRSR(11) = &HFE3F
220 GRCRSR(12) = &HFE3F
230 GRCRSR(13) = &HFE3F
240 GRCRSR(14) = &HFE3F
250 GRCRSR(15) = &HFFFF
255 REM XOR-Maske:
260 GRCRSR(16) = &H0000
270 GRCRSR(17) = &H0080
280 GRCRSR(18) = &H0080
290 GRCRSR(19) = &H0080
300 GRCRSR(20) = &H0080
310 GRCRSR(21) = &H0080
320 GRCRSR(22) = &H0080
330 GRCRSR(23) = &H3FFE
340 GRCRSR(24) = &H0080
350 GRCRSR(25) = &H0080
360 GRCRSR(26) = &H0080
370 GRCRSR(27) = &H0080
380 GRCRSR(28) = &H0080
390 GRCRSR(29) = &H0080
400 GRCRSR(30) = &H0000
410 GRCRSR(31) = &H0000
420 REGAX% = 9 : REM Funktionsnummer 9
430 REGBX% = 7 : REGCX% = 8 : REM Bezugspunkt im Zentrum des Fadenkreuzes
440 CALL MOUSE (REGAX%,REGBX%,REGCX%,GRCRSR(0)) : REM Aufruf des Treibers
450 REM Treiber-Existenz und Adresse wie in Bild 2 bestimmen!
```

Bild 4. So wird das Fadenkreuz in Basic programmiert

```
type Word = Integer; ( Ab Turbo Version 4.0 weglassen ! )
const Fadenkreuz: array[0..31] of Word =
                  ($FE3F,$FE3F,$FE3F,$FE3F,$FE3F,$FE3F,$8000,$8000,
                   $8000,$FE3F,$FE3F,$FE3F,$FE3F,$FE3F,$FE3F,$FFFF,
                   $0000,$0080,$0080,$0080,$0080,$0080,$0080,$3FFE,
                   $0080,$0080,$0080,$0080,$0080,$0080,$0000,$0000);

procedure SetzeFadenkreuz;
var ax,bx,cx,dx: Integer;
begin
  ax := 9;              ( Funktion 9 )
  bx := 7; cx := 8; ( Bezugspunkt )
  dx := ofs (Fadenkreuz[0]);
  Regs.es := seg (Fadenkreuz[0]); ( 5. Parameter für Funktion 9 )
  Mouse (ax,bx,cx,dx);
end;
  ...
  SetzeFadenkreuz;
  ...
```

Bild 5. Eine Turbo-Pascal-Prozedure zum Setzen des Fadenkreuzes von Bild 3

entspricht dann dem Drücken irgendeiner Maustaste.

Funktion 14: Light-Pen-Emulation ausschalten
Eingabe: AX = 14
.Rückgabewerte: keine.
Um einen Lichtgriffel gleichzeitig mit der Maus verwenden zu können, muß man, um die Griffelposition lesen zu können, zuerst die Emulation abschalten.

Funktion 15 : Empfindlichkeit der Maus einstellen
Eingabe: AX = 15, CX = Empfindlichkeit in X-Richtung in 1/100 Zoll pro 8 Pixel,
DX = Empfindlichkeit in Y-Richtung in 1/100 Zoll pro 8 Pixel.
Rückgabewerte: keine.
Mit dieser Funktion kann derjenige Weg eingestellt werden, um den man die Maus bewegen muß, um eine gewisse Verschiebung des Maus-Cursors zu erreichen; d. h. man muß die Maus um CX × 0.254 mm nach rechts bewegen, um die logische Maus-Cursor-X-Koordinate um 8 zu erhöhen, und um DX × 0,254 mm nach unten, um die Y-Koordinate um 8 zu erhöhen.

Funktion 16: Bildschirmfenster vor Überschreiben durch den Cursor schützen
Eingabe: AX = 16, DX = Zeiger zu einem 8-Byte-Daten-Block, der Position und Größe des zu schützenden Fensters beschreibt.
Rückgabewerte: keine.
Wenn auf einen gewissen Bildschirmbereich durch das Anwenderprogramm zugegriffen wird, muß der Maus-Cursor abgeschaltet werden, um bei einem Schreibzugriff zu vermeiden, daß der alte Bildinhalt zurückgespeichert wird (wie bei Funktion 9 erklärt), und um bei einem Lesezugriff nicht das Pixelmuster des Cursors mitzulesen. Das heißt, wenn ein Programm auf Bildinhalte zugreift, tritt ein störendes Flakkern des Cursors auf. Bequemer kann man durch Funktion 16 einfach das kritische Bildschirmfenster definieren, das vor Überschreiben durch den Cursor geschützt werden soll. In DX übergibt man einen Zeiger auf einen 8-Byte-Block, der folgende vier Integer-Parameter enthält:

1. Wort: logische X-Koordinate der Ecke links oben des geschützten Fensters
2. Wort: logische Y-Koordinate der Ecke links oben des geschützten Fensters
3. Wort: logische X-Koordinate der Ecke rechts unten des geschützten Fensters 4. Wort: logische Y-Koordinate der Ecke rechts unten des geschützten Fensters

In Pascal oder Basic entspricht diese Zeiger-Übergabe in DX der Übergabe des Bitmap-Zeigers bei Funktion 9, so daß Bild 3 zur Verdeutlichung herangezogen werden kann.

Funktion 18 : Aussehen eines großen Grafik-Cursors einstellen
Eingabe: AX = 18,
BL = X-Koordinate des Cursor-Bezugspunktes,
CL = Y-Koordinate des Cursor-Bezugspunktes,
BH = Breite des Cursors in Worten zu 16 Bit,
CH = Höhe des Cursors in Pixelzeilen,
DX = Zeiger auf die Cursor-Bitmap.
Rückgabewert: AX = -1 , falls diese Funktion unterstützt wird.
Das Erstellen der Bitmap erfolgt wie unter Funktion 9 beschrieben, nur muß die Breite und Höhe entsprechend beachtet werden. Die Breite kann nur in Vielfachen von 16 Bit eingestellt werden, die Höhe dage-

```
{ ***************************************** }
{ *       M a u s - T o o l b o x         * }
{ *             Version 1.1               * }
{ ***************************************** }
{ *  19.01.1988 by C. Brunner, München    * }
{ ***************************************** }
{ alle mit *UNIT* als Kommentar versehenen Zeilen nur einfügen, wenn }
{ die Maus-Toolbox als UNIT unter Turbo-Pascal 4.0 verwendet wird !! }

unit MouseBox; { *UNIT* }

interface      { *UNIT* }

uses dos; { nur ab Turbo 4.0 }

{ Folgende beide Typvereinbarungen nur bis Turbo Version 3.x erforderlich ! }
{type
  Word = Integer;
  Registers = record
                case Integer of
                  0: (AX,BX,CX,DX,BP,SI,DI,DS,ES,Flags: Word);
                  1: (AL,AH,BL,BH,CL,CH,DL,DH: Byte);
              end;
  }

type ButtonStateType = (Pressed,Released);  { mögl. Maustastenzustände }
     GraphCursorType  = record                  { Typ für Graphikcursor }
                          RefPointX: Integer;          { X-Bezugspunkt }
                          RefPointY: Integer;          { Y-Bezugspunkt }
                          Bitmap: array[0..31] of Word;    { Bitmuster }
                        end;

     SelType = (Hardware,Software);             { Arten von Textcursors }
     StateType = (On,Off);                 { Zustand von MouseCursor etc. }
     TextCursorType = record                          { Textcursor Typ }
                        Sel : SelType;        { Hardware oder Software }
                        case Boolean of
                          True : (Start,Stop: Integer);
                          False: (AndMask,XorMask: Word);
                      end;

const Left = 1;        { zum Aufruf von ButtonPressed() etc verwenden ! }
      Right = 2;
      Middle = 4;

      { Vordefinierte Graphik-Cursor-Records }
      { ------------------------------------ }
      Cross1: GraphCursorType = ( Fadenkreuz mit dunklem Rahmen )
             (RefPointX: 7;
              RefPointY: 8;
              Bitmap: ($FE3F,$FE3F,$FE3F,$FE3F,$FE3F,$FE3F,$8000,$8000,
                       $8000,$FE3F,$FE3F,$FE3F,$FE3F,$FE3F,$FE3F,$FFFF,
                       $0000,$0080,$0080,$0080,$0080,$0080,$0080,$3FFE,
                       $0080,$0080,$0080,$0080,$0080,$0080,$0000,$0000));
      Cross2: GraphCursorType = ( Fadenkreuz ohne Rahmen )
             (RefPointX: 7;
              RefPointY: 8;
              Bitmap: ($FFFF,$FFFF,$FFFF,$FFFF,$FFFF,$FFFF,$FFFF,$FFFF,
                       $FFFF,$FFFF,$FFFF,$FFFF,$FFFF,$FFFF,$FFFF,$FFFF,
                       $0000,$0080,$0080,$0080,$0080,$0080,$0080,$3FFE,
                       $0080,$0080,$0080,$0080,$0080,$0080,$0000,$0000));
      Arrow: GraphCursorType = ( der übliche Pfeil )

            (RefPointX: -1;
             RefPointY: -1;
             Bitmap: ($9FFF,$8FFF,$87FF,$83FF,$81FF,$80FF,$807F,$803F,
                      $801F,$800F,$80FF,$887F,$987F,$FC3F,$FC3F,$FEFF,
                      $0000,$2000,$3000,$3800,$3C00,$3E00,$3F00,$3F80,
                      $3FC0,$3FE0,$3E00,$2300,$0300,$0180,$0180,$0000));
```

Bild 6. Die Maus-Toolbox ist bereits für Turbo-Pascal 4.0 vorbereitet

```
      { Vordefinierte Text-Cursor-Records }
      Box: TextCursorType = ( viereckiger Block )
           (Sel: Hardware;  { Hardware-Cursor   }
            Start: 1;       { Startzeile        }
            Stop: 11);      { Endzeile          }

      HalfBox: TextCursorType = ( untere Hälfte des Blocks )
               (Sel: Hardware;
                Start: 5;
                Stop:  11);

      Underline: TextCursorType = ( doppelte Unterstreichung )
                 (Sel: Hardware;
                  Start: 11;
                  Stop:  12);

      Smily: TextCursorType = ( helles Gesicht          )
             (Sel: Software;   { diesmal Software-Cursor }
              AndMask: $0000;  { AND - Maske             }
              XorMask: $0F01); { XOR - Maske             }

var MouseCursorVisible: Boolean;    { Flag, ob Mauscursor gerade sichtbar }
{ Ab hier bis zum Wort "implementation" nur für *UNIT* - Erzeugung ! }
procedure Mouse(var M1,M2,M3,M4 : Integer);
procedure SetHercGraph(page: byte);
procedure NoHercGraph;
procedure MouseCursor (State: StateType);
procedure ToggleMouseCursor;
procedure ReadMouseStatus (var Buttons,XPos,YPos: Integer);
procedure SetMousePos (x,y: Integer);
procedure ButtonInfo (Button: Integer; State: ButtonStateType;
                      var ButtonSts, Count, X, Y : Integer);
procedure SetMouseXRange (Xmin,Xmax: Integer);
procedure SetMouseYRange (Ymin,Ymax: Integer);
procedure GraphCursor (Cursor: GraphCursorType);
procedure TextCursor (Crsr: TextCursorType);
procedure GetMouseCounters (X,Y : Integer);
procedure LightPenEmulation (State:StateType);
procedure MouseSensitivity (XSens,YSens: Integer);
function MouseDriverLoaded: Boolean; { liefert True, wenn Treiber geladen }
function MouseInit: Boolean;
function ButtonPressed(Button:Integer): Boolean;
function MousePosX: Integer;
function MousePosY: Integer;

implementation { *UNIT* }

const HercGM: byte = 0;   { enthält beim Herkules-Graphik-Mode die Pagenr.}
var Regs : Registers;     { dient der Registerübergabe beim Mausinterrupt }
    H1,H2,H3,H4: Integer; { Hilfsvariablen zum Aufruf des Maustreibers }


function MouseDriverLoaded: Boolean; { liefert True, wenn Treiber geladen }

const IntRet = 207;
var MouseOffset : Integer absolute 0:204; {  = $33*4 }
    MouseSegment: Integer absolute 0:206; {  = $33*4 + 2 }
begin
  If (MouseSegment <> 0) and
     (MouseOffset <> 0) and
     (Mem[MouseSegment:MouseOffset] <> IntRet)
     then MouseDriverLoaded := True
     else MouseDriverLoaded := False;
end;

procedure Mouse (var M1,M2,M3,M4 : Integer);
{ ruft den Maustreiber per INT 33h mit den 4 Integer-Parametern M1-4 auf }
```

```
begin
  with Regs do begin
    AX := M1;
    BX := M2;
    CX := M3;
    DX := M4;
    Intr(51,Regs);
    M1 := AX;
    M2 := BX;
    M3 := CX;
    M4 := DX;
  end;
end;

procedure SetHercGraph (page: byte);
{ muß vor MouseInit aufgerufen werden, wenn man eine Hercules-Karte }
{ im Grafikmodus verwendet; page=Grafikpage 0 oder 1               }
begin
  If page=0 then HercGM := 6
  else HercGM := 5;
end;

procedure NoHercGraph;
{ dient zum Abschalten des Hercules-Graphik-Modus des Maustreibers }
begin
  HercGM := 0;
end;

function MouseInit: boolean;
{ Initialisieren des Treibers, liefert true, wenn alles ok ist und false, }
{ wenn die Maus, der Treiber oder die serielle Karte nicht gefunden wurde }
begin
  MouseCursorVisible := False;
  If MouseDriverLoaded Then begin
    If HercGM<>0 then Mem[$40:$49] := HercGM; { Herkuleskarte ? }
    H1:=0;                                    { Maustreiber Funktion 0 }
    Mouse (H1,H2,H3,H4);
    MouseInit := (H1 = -1);                   { -1 heißt alles ok }
  end
  else MouseInit := False;
end;

procedure MouseCursor (State: StateType);
{ schaltet den Text- oder Graphikcursor ein bzw. aus        }
{ (Art des Cursors abhängig vom aktuellen Bildschirmmodus)  }
begin
  If State = On Then begin
                       H1 := 1; { Maustreiber Funktion 1 }
                       MouseCursorVisible := False;
                     end
                Else begin
                       H1 := 2; { Maustreiber Funktion 2 }
                       MouseCursorVisible := False;
                     end;
  Mouse (H1,H2,H3,H4);
end;

procedure ToggleMouseCursor;
{ kehrt den Sichtbarkeits-Zustand des Mauscursors um                }
{ wird benötigt, um den Cursor durch SetMousePosition sofort sichtbar }
{ positionieren zu können                                           }
begin
  If MouseCursorVisible then MouseCursor(Off)
  else MouseCursor(On);
end;

procedure ReadMouseStatus (var Buttons,XPos,YPos: Integer);
{ liest aktuellen Maustastenstatus, X- und Y-Position ein           }
{ XPos und YPos beziehen sich immer auf das logische Koordinatensystem }
{ Maustastenstatus: Bit 0,1,2 entspr. linke,rechte,mittlere Taste   }
{ log. 1 heißt dabei "gedrückt", log. 0 heißt "nicht gedrückt"      }
{ Abfrage, ob gedrückt also z.B. If (Buttons and Middle)<>0 Then ...  }
begin
  H1 := 3; { Maustreiber Funktion 3 }
  Mouse (H1,H2,H3,H4);
  Buttons := H2;
  XPos := H3;
  YPos := H4;
end;

function ButtonPressed (Button:Integer): boolean;
{ liefert true, wenn die linke Maustaste gedrückt ist }
{ Aufruf z.B. "If ButtonPressed (Left) Then ...", Left,Right,Middle sind mögl.}
begin
  H1 := 3; { Maustreiber Funktion 3 }
  Mouse (H1,H2,H3,H4);
  ButtonPressed := (H2 and Button)<>0;
end;

function MousePosX: Integer;
{ liefert die aktuelle X-Position der Maus }
begin
  H1 := 3; { Maustreiber Funktion 3 }
  Mouse (H1,H2,H3,H4);
  MousePosX := H3;
end;

function MousePosY: Integer;
{ liefert die aktuelle Y-Position der Maus }
begin
  H1 := 3; { Maustreiber Funktion 3 }
  Mouse (H1,H2,H3,H4);
  MousePosY := H4;
end;

procedure SetMousePos (X,Y: Integer);
{ setzt den Mauscursor an die angegebene Stelle (x,y) }
begin
  H1 := 4; { Maustreiber Funktion 4 }
  Mouse (H1,H2,X,Y);
  ToggleMouseCursor;    { Trick, um die Positionsänderung bei eingeschaltetem }
  ToggleMouseCursor;    { Cursor sofort zu bewirken                           }
end;

procedure ButtonInfo (Button: Integer;     { links,rechts,Mitte entspr. 1,2,4 }
                                           { Left/Right/Middle verwenden !    }
                      State: ButtonStateType;     { Released/Pressed          }
                      var ButtonSts,              { Tastenstatus              }
                          Count,                  { Anzahl Releases/Presses   }
                          X,Y:Integer);           { X/Y-Cursorposition        }
{ übergibt folgende Informationen bezüglich Button (aus Left,Right,Middle)    }
{                                  und State (aus Released,Pressed            }
{ ButtonStst : spiegelt den momentanen Tastenzustand wieder s. ReadMouseStatus}
{ Count      : Anzahl, wie oft seit letztem Aufruf von ButtonInfo die Taste   }
{              Button gedrückt/losgelassen wurde (je nach State)              }
{ X,Y        : log. X/Y-Koordinaten des Cursors be letztem Drücken/Loslassen  }
begin
  If State=Pressed then H1 := 5 else H1 := 6;     { Press oder Release-Info ? }
                                                  { entspr. Funktion 5 bzw. 6 }
  H2 := Button -1; If H2 = 3 Then H2 := 2;                { Tastennummer      }
  Mouse (H1,H2,X,Y);
  ButtonSts := H1;
  Count := H2;
end;

procedure SetMouseXRange (Xmin,Xmax: Integer);
{ dient dem Einstellen der minimalen / maximalen horizontalen Cursorposition }
begin
  H1 := 7; { Maustreiber Funktion 7 }
  Mouse (H1,H2,Xmin,Xmax);
  ToggleMouseCursor; ToggleMouseCursor;
end;
```

```
procedure SetMouseYRange (Ymin,Ymax: Integer);
{ dient dem Einstellen der minimalen / maximalen vertikalen Cursorposition }
begin
  Hi := 8; { Maustreiber Funktion 8 }
  Mouse (Hi,H2,Ymin,Ymax);
  ToggleMouseCursor; ToggleMouseCursor;
end;

procedure GraphCursor (Cursor: GraphCursorType);
{ Aussehen des Graphikcursors einstellen; siehe Erläuterung der Bitmap }
{ und Erklärung bei den Typvereinbarungen                               }
begin
  Hi := 9; { Maustreiber Funktion 9 }
  H2 := Cursor.RefPointX;
  H3 := Cursor.RefPointY;
  H4 := ofs (Cursor.Bitmap[0]);
  Regs.ES := seg (Cursor.Bitmap[0]);
  Mouse (Hi,H2,H3,H4);
end;

procedure TextCursor (Crsr: TextCursorType);
{ Selektiert den Harware-Text-Cursor; z.B. Underline,Box,HalfBox,Smily }
begin
  Hi := 10; { Maustreiber Funktion 10 }
    if Crsr.Sel = Hardware Then H2 := 1 else H2 := 0;
    H3 := Crsr.Start; { Startzeile bzw. AND-Maske }
    H4 := Crsr.Stop;  { Endzeile bzw. XOR-Maske   }
    Mouse (Hi,H2,H3,H4);
end;

procedure GetMouseCounters (X,Y : Integer);
{ liefert die Veränderung der Mausposition seit dem letzten Aufruf }
begin
  Hi := 11;
  Mouse (Hi,H2,X,Y);
end;

procedure LightPenEmulation (State:StateType);
{ schaltet Lichtgriffelemulationsmodus ein }
begin
  if State = On Then Hi := 13 { Maustreiber Funktion 13 }
                else Hi := 14; { Maustreiber Funktion 14 }
  Mouse (Hi,H2,H3,H4);
end;
```

```
procedure MouseSensitivity (XSens,YSens: Integer);
{ Zum Einstellen der Empfindlichkeit der Maus; Default: YSens=16, XSens=8 }
{ XSens/YSens gibt an, wieviele 1/100 Zoll die Maus bewegt werden muß,    }
{ um den Mauscursor um 8 Pixels in X/Y-Richtung zu bewegen                }
begin
  Hi := 15; { Maustreiber Funktion 15 }
  Mouse (Hi,H2,XSens,YSens);
end;

end.  { *UNIT* }
```

```
Warten, bis die linke Taste gedrückt und wieder losgelassen wurde und dann die
Position des Mauscursors feststellen:

repeat until ButtonPressed (Left);
repeat until not ButtonPressed (Left);
A := MousePosX; B := MousePosY;

Fehlermeldung ausgeben, falls Mausdriver nicht geladen:

If not MouseDriverLoaded Then
  begin Write ('Maustreiber nicht geladen!');
  Halt;
  end;

Grafik- oder Textcursor ändern:

GraphCursor (Crossl); {=Fadenkreuz oder TextCursor (Smily); Gesicht}

Maustreiber für Hercules-Karte Seite 0 initialisieren und Cursor einschalten:

... Grafikmodus einschalten...
SetHercGraph (0); { Seite 0 }
if not MouseInit Then Error; { Fehler, daher z.B. Prozedur Error aufrufen }
MouseCursor (On);
```

Bild 7. Beispiele zur Anwendung der Toolbox

gen in einzelnen Zeilen. Um das Beispiel von Bild 3 mit dieser Funktion anzuwenden, müssen die Register wie folgt geladen werden:

AX = 18, BL = 255 (=-1), CL = 255 (= -1), BH = 1, CH = 16, DX = Adresse der Bitmap.

Folgende Beziehung begrenzt die maximale Größe des Cursors:
$((2 \times BH) + 1) \times CH < 145$.

Um festzustellen, ob Funktion 18 überhaupt existiert, kann man nach dem Aufruf den Inhalt von AX prüfen; -1 bedeutet „ja", alles andere „nein".

Die Maus-Toolbox

Bild 6 zeigt eine kleine Turbo-Pascal-Toolbox, mit welcher die Maus schon recht komfortabel bedient werden kann. Ziel war es, nicht immer die Funktionsnummern und Parameterreihenfolgen des Maustreiberaufrufes nachblättern zu müssen.

Turbo-Pascal unterstützt ab der Version 4.0 sogenannte Units. Dabei handelt es sich um bereits compilierte Unterprogrammpakete, von denen nur die im Hauptprogramm wirklich benötigten Routinen miteingebunden werden. Somit muß man weder bei jeder Änderung des Hauptprogrammes den gesamten Quelltext neu übersetzen, noch werden die erzeugten COM- bzw. EXE-Dateien durch unnötigen Code aufgebläht (typisches Beispiel: Verwendung der Grafik-Toolbox unter Turbo 3.x nur zum Ein- und Ausschalten des Grafikmodus). Will man also eine MausBox-Unit erzeugen, kann man das abgedruckte Listing ohne Änderungen übernehmen. Ist dies nicht der Fall, sind alle mit dem Wort *UNIT* im Kommentar gekennzeichneten Zeilen zu entfernen. Bei Turbo-Versionen früher als 4.0 müssen außerdem die beiden im Listing „wegkommentierten" Typdeklarationen eingefügt werden. Da die Funktionsweise der einzelnen Routinen ohne Probleme aus dem Programm entnommen werden kann, sollen hier zum Abschluß nur noch einige typische Aufrufbeispiele gezeigt werden.
Die Befehle, mit denen das System auf das Drücken der mittleren Maus-Taste wartet lauten:
repeat until ButtonPressed(Right);
Weitere Tips sind in *Bild 7* dargestellt. Man muß nur noch beachten, daß die Hercules-Karte eine Sonderbehandlung per SetHercGraph () benötigt. Alle anderen üblichen Grafikkarten werden automatisch erkannt. □

Suchen sie einen Partner mit dem Sie wachsen können?

Die INTRA-führender Hersteller von Monitoren - bietet seit Jahren Ihren Kunden eine stetig wachsende Anzahl von Anzeigegeräten und Displays.
Von der Bildröhre bis zum 20" Farb-Videospielmonitor, vom Composite-Monitor bis zum IBM-kompatiblen Monochrom-Monitor und 14" CGA/EGA Multi-SYNC-Bildschirm.
Wir haben nicht nur das Wissen und Können, sondern auch die Qualität.
Wir wollen expandieren und weiterhin qualitativ hochwertige HiTech entwickeln - schon heute fertigen wir bis zu 30,000 Bildschirme monatlich.
Mit 3, 5 Mio Dollar Kapital, 250 Mitarbeitern und einem Werksgelände von knapp 6,000 qm haben wir die besten Voraussetzungen für Wachstum.
Unsere Zukunftspläne erfordern eine verstärkte Präsenz in den Bereichen Ganzseitenmonitor sowie eine Ausweitung der Desk-top-, Modem- und Lap-top-Technologien.
Sie sind in diesem Bereich tätig, verfügen über ausreichende Erfahrung und wollen kooperieren?
Dann sollten Sie mit uns Kontakt aufnehmen. Es wird sich für Sie lohnen.

"VGA" MONOCHROME MONITOR
MODEL 14HP34V
1. 14" FLAT SCREEN
2. PS/2, VGA COMPATIBLE (31.5KHZ) ANALOG VIDEO
3. CRT: PAPER-WHITE, AMBER,GREEN

14" MONOCHROME MONITOR
MODEL 14HP33T
1. 14" FLAT SCREEN
2. DUAL FREQ (15.75/18.432KHz)
3. CRT: PAPER-WHITE, AMBER,GREEN
4. WITH REVERSE SWITCH

14" HIGH-RESOLUTION COLOR MONITOR
MODEL
14CH113 (EGA 640 × 350)
14CH114 (CGA 640 × 200)
14CH115 (MULTI-SYNC 800 × 600)
14CH116 (VGA 640 × 350
640 × 400
640 × 480)

• "VGA" MONOCHROME MONITOR

• 14" MONOCHROME MONITOR

• 14" HIGH-RESOLUTION COLOR MONITOR

INTRA

Intra Electronics Co., Ltd.
Room No. 618, 6th Fl., 9, Lane 3,
Min Sheng West Rd., Taipei, Taiwan, R.O.C.
Tel: (02)597-7027 Tlx: 19925 INTRA Fax: 886-2-5418513

IBM, EGA, and CGA are registered tardemarks of the International Business Machines Corp.

VED. FCC APPROVED
• OEMs WELCOME !!

Robert Fehrmann

Dynamische Arrays in Turbo-Pascal

Bei Turbo-Pascal ist die Verwendung von variablen Arraygrenzen nicht vorgesehen. Die Größe eines Feldes wird normalerweise im Programm festgelegt und kann zur Laufzeit nicht verändert werden. Mit einem kleinen Trick ist es möglich, Felder zu erzeugen, deren Größe erst zur Laufzeit festgelegt wird.

Ein weiterer Vorteil bei der hier geschilderten Methode ist, daß die Felder keinen Platz in dem (bei Version 3 des Compilers) auf 64 KByte begrenzten Code- oder Daten-Segment belegen. Voraussetzung bei der Verwendung dieses Tricks ist die Kenntnis darüber, wieviel Platz von den einzelnen Komponenten des Feldes belegt wird (*Tabelle*). Diese Information läßt sich dem Handbuch entnehmen. Das folgende Beispiel zeigt die Vorgehensweise an Integer- und Real-Arrays.

Definitionen

Zuerst wird ein Aufzählungstyp Type_Def definiert, der all jene Typen enthält, zu denen Arrays mit variablen Grenzen erzeugt werden sollen (*Bild 1*). In der gleichen Reihenfolge muß nun im Feld Type_Size die Speicherplatzbelegung für die vorher definierten Typen eingetragen werden. Diese Definitionen sind nicht unbedingt notwendig, machen das Programm aber durchschaubarer und vor allem auch sicherer, da die Prozedur Freemem recht unangenehm auf falsche Parameter reagiert.
Als nächstes werden für alle verwendeten Typen (siehe Typ_Def) Dummy-Array-Typen sowie Zeiger auf diese Array-Typen definiert. Im Beispiel von Bild 1 geschieht dies in den Zeilen 23...29.

Zugriffsprozeduren

Die beiden Prozeduren New_Array und Dispose_Array verwalten dynamisch den Speicher für die Arrays. Sie dienen dem Erzeugen und Löschen der dynamischen Arrays. Der erste Parameter der Prozedur ist immer der Zeiger auf das gewünschte Feld. Damit der Compiler keinen Typenfehler meldet, wird hier ein nichttypisierter Variablenparameter verwendet. Die Turbo-Prozeduren Getmem und Freemem akzeptieren dagegen nur typisierte Parameter.
Durch die Absolut-Deklaration von Any_Pointer, der so mit dem Parameter von New_Array (Dispose_Array) identisch gemacht wird, kann dann die Turbo-eigene Speicherverwaltung angesprochen werden. Der zweite Parameter der neuen Zugriffsprozeduren enthält den Typ des dynamischen Arrays (aus Typ_Def). Der dritte Parameter legt die Anzahl der Array-Komponenten fest. Es ist klar, daß dieser Parameter immer größer als Null sein muß.
Innerhalb der Prozedur, wird aus der Typ-Information in Type_Size und dem dritten Parameter die Anzahl der zu belegenden Bytes. Da über Pointer gearbeitet wird, kann der gesamte verbleibende Speicher

```
 1:  0 (*************************************************************************)
 2:  0 (*                                                                       *)
 3:  0 (*         Include_Datei zur Erzeugung von Integer/Real Arrays           *)
 4:  0 (*                        mit variablen Grenzen                          *)
 5:  0 (*                                                                       *)
 6:  0 (*************************************************************************)
 7:  0
 8:  0 (*=======================================================================*)
 9:  0 (* In Type_Def werden alle Typedefinitionen aufgeführt, zu denen mit     *)
10:  0 (* der Prozedur New_Array ein Feld beliebiger Größe erzeugt werden soll. *)
11:  0 (* In Type_Size muß der zu diesem Type benötigte Speicherbedarf(in BYTE) *)
12:  0 (* angegeben werden. (Reihenfolge beachten!)                             *)
13:  0 (*-----------------------------------------------------------------------*)
14:  0
15:  0 TYPE  Type_Def                          = (Integer_Type,Real_Type);
16:  0
17:  0 CONST Type_Size : ARRAY[0..1] OF Integer = (     2     ,     6   );
18:  0
19:  0 (*=======================================================================*)
20:  0 (* Array- und Pointerdefinitionen zu den Typen aus Type-Def.              *)
21:  0 (*-----------------------------------------------------------------------*)
22:  0
23:  0 TYPE Integer_Array = ARRAY[1..1] OF Integer;
24:  0
25:  0      Real_Array    = ARRAY[1..1] OF Real;
26:  0
27:  0      Pointer_To_Integer_Array = ^Integer_Array;
28:  0
29:  0      Pointer_To_Real_Array    = ^Real_Array;
30:  0
31:  0 (*=======================================================================*)
32:  0 (* Mit den Proceduren New_Array bzw. Dispose_Array werden Die Felder er- *)
33:  0 (* zeugt bzw. gelöscht. Die Prozeduren erhalten als Parameter einen      *)
34:  0 (* Pointer, welcher auf das zu erzeugende Feld zeigen soll, sowie den    *)
35:  0 (* Typ und die Anzahl der Elemente.                                      *)
36:  0 (* Achtung : Wird ein Feld mit Dispose_Array gelöscht, so muß Array_Size *)
37:  0 (*           genau den Wert haben, den Array_Size bei der Erzeugung mit  *)
38:  0 (*           New_Array hatte!                                            *)
39:  0 (*-----------------------------------------------------------------------*)
40:  0
41:  0 PROCEDURE New_Array(VAR Pointer;Element_Type:Type_Def;Array_Size:Integer);
42:  0
43:  0 VAR Any_Pointer : ^Integer ABSOLUTE Pointer;
44:  0
45:  0 BEGIN
46:  1 Getmem(Any_Pointer,Type_Size[Integer(Element_Type)]*Array_Size)
47:  1 END;
48:  0
49:  0 PROCEDURE Dispose_Array(VAR Pointer;Element_Type:Type_Def;Array_Size:integer);
50:  0
51:  0 VAR Any_Pointer : ^Integer ABSOLUTE Pointer;
52:  0
53:  0 BEGIN
54:  1 Freemem(Any_Pointer,Type_Size[Integer(Element_Type)]*Array_Size);
55:  1 END;
56:  0
57:  0 (*=======================================================================*)
```

Bild 1. Diese Include-Datei enthält die Vereinbarungen für dynamische Arrays

Speicherplatzbelegung der wichtigsten Typen von Turbo-Pascal

Typ	belegter Speicherplatz
Char	1 Byte
Byte	1 Byte
Integer	2 Bytes
Real	6 Bytes
String[x]	x+1 Byte

angesprochen werden. So würde der Aufruf New_Array (Array1, Integer_Type, 100) einen Zeiger auf ein Integer-Feld mit 100 Komponenten bereitstellen.

Wichtig ist, daß der Lösch-Aufruf mit Dispose_Array dieselben Parameter haben muß, wie der korrespondierende Aufruf von New_Array.

Beispiel

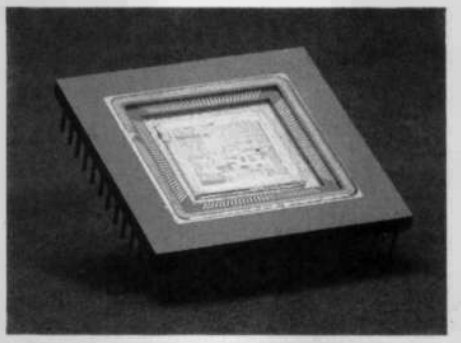
Bild 2 zeigt ein Programm, das die Anwendung der beiden Zugriffsprozeduren demonstriert. Die einzige Änderung gegenüber der Verwendung normaler Arrays ist die Dereferenzierung (es werden ja Pointer verwendet). Statt Array1[i] heißt es nun Array1 ^ [i].

```
1:  0 {$I C:Dynarray.Pas}
2:  0
3:  0 {##########################################################################}
4:  0 {# In diesem Programm werden ein Integer- und ein Realfeld erzeugt.      #}
5:  0 {# Die Größe der Felder wird allerdings erst zur Laufzeit des Programms  #}
6:  0 {# vom Benutzer eingegeben.                                              #}
7:  0 {##########################################################################}
8:  0
9:  0 VAR Array1 : Pointer_To_Integer_Array;
10: 0     Array2 : Pointer_To_Real_Array;
11: 0     max,i  : Integer;
12: 0
13: 0 BEGIN
14: 1 clrscr;
15: 1 write('Geben Sie bitte die Anzahl der Elemente ein : ');
16: 1 readln(max);
17: 1 New_Array(Array1,Integer_Type,max);
18: 1 New_Array(Array2,Real_Type,max);
19: 1 FOR i := 1 TO max
20: 1 DO BEGIN
21: 2    Array1^[i] := i;
22: 2    Array2^[i] := i;
23: 2    writeln(Array1^[i]:4,Array2^[i]:7:2);
24: 2    END;
25: 1 Dispose_Array(Array1,Integer_Type,max);
26: 1 Dispose_Array(Array2,Real_Type,max);
27: 1 END.
```

Bild 2. Ein Beispiel zum Testen der Routinen aus Bild 1

Bei diesem Beispiel wurde davon ausgegangen, daß die erste Komponente des dynamischen Feldes Array ^ [1] ist. Das rührt von der Definition in Bild 1 her. Stände dort zum Beispiel ARRAY[0..0] of Integer, hätten wir 0 als untere Grenze.

Schneller Coprozessor von Weitek



Abacus, der 121-Pin-Coprozessor von Weitek

Unter dem Namen Abacus (gelegentlich auch noch Weitek 3167 genannt) stellte Weitek einen numerischen Coprozessor in einer 20-MHz- und einer 25-MHz-Version für die 386er-PCs vor. Es handelt sich im Gegensatz zu seinem Vorgänger Weitek 1167 (drei Chips) um eine Single-Chip-Version. Sein entscheidender Vorteil ist eine gesteigerte Geschwindigkeit, die nach Angaben von Weitek rechenintensive Anwendungen um den Faktor 2 bis 5 schneller machen kann.

Für den Abacus ist ein Sockel in EMC-Norm (Erweiterter Mathematischer Coprozessor) erforderlich, in den auch Intels 80387-Coprozessor paßt. Ein EMC-Sockel ist in den meisten PCs mit einer 80386-CPU vorhanden. Mit einem Tochterboard von Weitek ist es möglich, daß sowohl der Intel- als auch der Weitek-Coprozessor installiert sind. Der Deskpro 386 in der 20-MHz- und in der 25-MHz-Version besitzt bereits standardmäßig zwei Sockel zur Aufnahme beider Coprozessoren.

Abacus enthält ein 32x32-bit-Register-File, eine 64-bit-ALU, einen Multiplizierer, einen Dividierer und eine Logik zur Berechnung von Wurzeloperationen; er wird in 1,5-Mikron-CMOS-Technologie hergestellt. Zur Zeit setzen insgesamt 17 Anwendungen den Abacus zur Geschwindigkeitssteigerung ein. C-, Fortran- und Pascal-Compiler unter DOS-Real-Mode, DOS-Protected-Mode und Unix-V.3-Systemen bieten Lahey Computer Systems, Metaware, Microway Inc., Green Hills, SAIC, OASYS und Silicon Valley Software an.

Der 5000-Stück-OEM-Preis des 20-MHz-Abacus beträgt ca. 450 Dollar, der 25-MHz-Version ca. 900 Dollar und des Tochterboards ca. 600 Dollar.

Mauro Bonomi, der Weitek-Produktmanager für Coprozessoren, wies auf die Tendenz hin, daß rechenintensive Programme innerhalb der PC-Software-Palette einen immer größeren Raum einnehmen. Gleichzeitig werde die Entwicklung bei den numerischen Coprozessoren MFLOPS (million floating-point-operations per second) bald billiger machen als MIPS (million instructions per second).

Von dem neuen Coprozessor können vor allem technisch-wissenschaftliche Anwendungen profitieren, aber auch Software aus dem Geschäftsbereich. Sehr rechenintensiv (mit Fließkommaarithmetik) sind zum Beispiel elektrische und mechanische Simulationen, statistische Analysen und Tabellenkalkulationen, ganz besonders bewegte komplexe Grafik.

Damit sind PCs jetzt noch eher in der Lage, in eine ursprüngliche Domäne der Mainframes, Minicomputer und Workstations einzudringen. Weitek gibt an, daß ein PC ausgestattet mit Abacus die numerische Leistungsfähigkeit einer VAX 8650 erreicht. *br*

Günther Sternberg

Grafik- oder Textmodus

Betriebsart bei Hercules-kompatiblen Karten einfach festgestellt

Residente Programme mit Bildschirmausgabe müssen beim Aufruf prüfen, ob sich die aktive Bildschirmkarte im Text- oder Grafikmodus befindet. Bei Hercules-kompatiblen Karten ist diese Unterscheidung nicht so einfach. Eine scheinbar unwichtige Funktion birgt die Lösung des Problems.

Bei der Hercules-Monochrom-Grafikkarte erfolgt im Gegensatz zu Grafikkarten nach dem CGA-, EGA- und VGA-Standard die Umschaltung zwischen Text- und Grafikmodus nicht über den Video-Interrupt 10H. Der Grund hierfür ist sehr einfach. IBM bot in den Anfangsjahren des PCs nur zwei Bildschirmkarten an, den nichtgrafikfähigen Monochrom-Display-Adapter (MDA) und den Color-Grafik-Adapter (CGA). Letzterer konnte sich in Europa wegen seiner geringen Auflösung nicht richtig durchsetzen. Der Hersteller der echten Hercules-Karte machte aus der Not eine Tugend und entwickelte eine Karte, die sich im Textmodus wie die MDA verhält und zusätzlich einen Grafikmodus mit einer Auflösung von 720 × 348 Punkten bietet. Obwohl die Hercules-Karte bzw. die Hercules-kompatiblen-Karten heute weltweit die wohl am häufigsten eingesetzten Grafikkarten sind, hat IBM diese Karte ignoriert, und das BIOS jedes IBM-kompatiblen PCs betrachtet eine Hercules-Karte nach wie vor als eine MDA-Karte. Daher liefert die Funktion 15H (Current Video State) des Interrupts 10H immer den Anzeigemodus 7 zurück, egal ob sich die Hercules-Karte im Text- oder im Grafikmodus befindet. Wegen der fehlenden BIOS-Unterstützung muß ein Anwenderprogramm den Wechsel von Text auf Grafik und umgekehrt durch umprogrammieren mehrerer Register selbst vornehmen. Die Unterscheidung zwischen Text- und Grafikmodus muß daher durch Analyse der Register der Hercules-Karte erfolgen. Beim Display-Mode-Control-Port (E/A-Adresse 03B8H) ist zwar das Bit 1 für die Unterscheidung von Text- und Grafik zuständig, doch kann dieses Register nur geschrieben und nicht gelesen werden. Zur Unterscheidung der Betriebsart gibt es kein spezielles Register. Ein Trick, der auch bei den meisten Hercules-kompatiblen Grafikkarten funktioniert, hilft hier weiter.

Da der 6845-Grafik-Controller der Hercules-Karte intern bereits eine komplette Logik zur Unterstützung eines Lichtgriffels (Lightpen) aufweist und nur wenige externe Bauelemente nötig sind, enthält jede Hercules-Karte standardmäßig eine Lichtgriffel-Schnittstelle. Da Lichtgriffel im Zeitalter der Maus heute fast nicht mehr eingesetzt werden, sind an dieser Stelle wohl einige Worte zur Funktionweise angebracht. Ein Lichtgriffel enthält an seiner Spitze eine hochempfindliche Fotodiode. Beim direkten Kontakt des Lichtgriffels mit dem Bildschirm reagiert die Fotodiode auf den vorbeilaufenden Elektronenstrahl des Monitors und erzeugt einen kleinen elektrischen Impuls. Sobald der 6845-Prozessor an einem bestimmten Eingang diesen Impuls erkennt, speichert er die aktuelle Position des Elektronenstrahls in einem speziellen Registerpaar ab. Die Position des Elektronenstrahls besteht aus der Nummer des zum Zeitpunkt des Impulses gerade ausgegebenen Textzeichens. Alle Textzeichen werden von links nach rechts und von oben nach unten fortlaufend durchnummeriert.

```
                .8086
cseg            segment  byte public
                assume   cs:cseg, ds:cseg, es:cseg
                public   HGC_Mode
                                                ;
; ------------------------------------------------------------------------
; Diese Funktion ermöglicht bei einer Hercules-Monochrom-Grafikkarte die
; Unterscheidung von Text- und Grafikmodus. Die Routine ist für einen
; Aufruf durch Turbo-Pascal Version 4.0 ausgelegt. Bei Turbo-Pascal bis
; Version 3.0 ist das Funktionsergebnis mittels dem Zero-Flag zu übergeben
; und beim Rücksprung ein Wort vom Stack zu entfernen (Ret 1).
;
; Turbo-Funktionskopf:   Function HGC_Mode : Boolean;
;
; (C) Günther Sternberg 1988                              Stand 18.4.1988
; ------------------------------------------------------------------------
                                                ;
Tests           equ      2000                   ; Anzahl der Testdurchläufe
Threshold       equ      3000                   ; Grenzwert für Lightpen-
                                                ; Zähler zur Unterscheidung
                                                ; von Text- und Grafikmodus
                                                ;
HGC_Mode        proc     near                   ; Prozedur mit Near-Return
                push     bp                     ; BP-Register sichern
                mov      bp, sp                 ; SP-Register nach BP kopieren
                mov      bx, 0                  ; Vorgabe für max. Lightpen-
                                                ; Position
                mov      cx, Tests              ; Anzahl der Durchläufe
                                                ;
L001:           mov      dx, 03BBH              ; Adresse des Registers zum
                                                ; Löschen des LP-Flip-Flops
                out      dx, al                 ; Lightpen-Flip-Flop löschen
                mov      dx, 03B9H              ; Adresse des Registers zum
                                                ; Setzen des LP-Flip-Flops
                out      dx, al                 ; Lightpen-Flip-Flop setzen
                mov      dx, 03B4H              ; Adresse des Indexregisters
                mov      al, 16                 ; Index des Lightpen-Registers
                out      dx, al                 ; Indexregister programmieren
                inc      dx                     ; Adresse des Datenregisters
                in       al, dx                 ; hi-Byte des Lightpen-Zählers
                mov      ah, al                 ; hi-Byte nach AH kopieren
                dec      dx                     ; Adresse des Indexregisters
                mov      al, 17                 ; Index des Lightpen-Registers
                out      dx, al                 ; Indexregister programmieren
                inc      dx                     ; Adresse des Datenregisters
                in       al, dx                 ; lo-Byte des Lightpen-Zählers
                cmp      ax, bx                 ; gelesene Lightpen-Position >
                                                ; bisherigem Maximum ?
                jle      L002                   ; kein neues Maximum
                mov      bx, ax                 ; neues Maximum speichern
L002:           loop     L001                   ; weiterer Durchlauf ?
                                                ;
                cmp      bx, Threshold          ; Grafik- oder Textmodus ?
                jle      L003                   ; Hercules-Karte im Textmodus
                mov      ax, 1                  ; Funktionsergebnis = True
                jmp      short L004             ; Ende der Funktion
L003:           xor      ax, ax                 ; Funktionsergebnis = False
L004:           mov      sp, bp                 ; BP-Register nach SP kopieren
                pop      bp                     ; BP-Register restaurieren
                ret                             ; Funktionsergebnis in AL
HGC_Mode        endp                            ;

cseg            ends
                end
```

Bild 1. Die scheinbar veraltete und unnötige Lichtgriffel-Steuerung dient als Retter in der Not

```
                .8086

cseg            segment byte public
                assume  cs:cseg, ds:cseg, es:cseg
                public  Enter_Graphic
                public  Leave_Graphic
                                                ;
Base            equ     0B000H                  ; Basis-Segment der Grafikkarte
                                                ;
Enter_Graphic   proc    near                    ; Prozedur mit Near-Return deklarieren
                push    bp                      ; BP-Register sichern
                push    ds                      ; DS-Register sichern
                mov     bp, sp                  ; SP-Register nach BP kopieren
                cld                             ; Direction-Flag löschen
                                                ;
                                                ; Register der Hercules-Karte mit den
                                                ; Werten für den Grapfikmodus laden
                                                ;
                mov     dx, 03BFH               ; Adresse des Configuration Switchs
                mov     al, 1                   ; Grafikmodus zulassen
                out     dx, al                  ; Configuration-Switch setzen
                mov     dx, 03B8H               ; Adresse des Display Mode Ctr. Ports
                mov     al, 02H                 ; Bildschirm dunkel schalten
                out     dx, al                  ; Display Mode Control Port setzen
                mov     ax, cs                  ; CS-Register nach AX kopieren
                mov     ds, ax                  ; CS-Register nach DS kopieren
                mov     si, Offset Graph_Data   ; Offset der Parameter-Tabelle laden
                mov     dx, 03B4H               ; Adresse des Index-Registers
                mov     bl, 0                   ; Startwert für Regsterindex
                mov     cx, 12                  ; Zähler für 12 Parameter
L015:           mov     al, bl                  ; Index des Registers laden
                out     dx, al                  ; Indexregister programmieren
                inc     dx                      ; Adresse des Data-Registers jetzt in DX
                lodsb                           ; Wert für Register laden
                out     dx, al                  ; Register programmieren
                dec     dx                      ; Adresse des Index-Registers jetzt in DX
                inc     bl                      ; Index des nächsten Registers
                loop    L015                    ; nächstes Register programmieren
                mov     dx, 03B8H               ; Adresse des Display Mode Ctr. Ports
                mov     al, 0AH                 ; Bildschirm wieder einschalten
                out     dx, al                  ; Display Mode Control Port setzen
                                                ;
                                                ; Grafik-Bildschirm löschen
                                                ;
                mov     ax, Base                ; Segment der Grafikkarte
                mov     es, ax                  ; Segment ins ES-Register kopieren
                mov     di, 0                   ; Startoffset des Bildspeichers
                mov     cx, 16384               ; Zahl der zu schreibenden Worte
                mov     ax, 0                   ; Wert für Bildspeicher
                rep     stosw                   ; Bildspeicher löschen
                                                ;
                mov     sp, bp                  ; BP-Register nach SP kopieren
                pop     ds                      ; DS-Register restaurieren
                pop     bp                      ; BP-Register restaurieren
                ret                             ; Ende der Prozedur
                                                ;
Graph_Data      db      35H, 2DH, 2EH, 07H, 5BH, 02H ; Registerwerte für Grafikmodus
                db      58H, 58H, 02H, 03H, 00H, 00H ;
Enter_Graphic   endp                            ;
                                                ;
Leave_Graphic   proc    near                    ; Prozedur mit Near-Return deklarieren
                push    bp                      ; BP-Register sichern
                push    ds                      ; DS-Register sichern
                mov     bp, sp                  ; SP-Register nach BP kopieren
                cld                             ; Direction-Flag löschen
                                                ;
                                                ; Register der Hercules-Karte mit den
                                                ; Werten für den Textmodus laden
                                                ;
                mov     dx, 03BFH               ; Adresse des Configuration Switchs
                mov     al, 0                   ; Grafikmodus sperren
                out     dx, al                  ; Configuration-Switch setzen
                mov     dx, 03B8H               ; Adresse des Display Mode Ctr. Ports
                mov     al, 20H                 ; Bildschirm dunkel schalten
                out     dx, al                  ; Display Mode Control Port setzen
                mov     ax, cs                  ; CS-Register nach AX kopieren
                mov     ds, ax                  ; CS-Register nach DS kopieren
                mov     si, Offset Text_Data    ; Offset der Parameter-Tabelle laden
                mov     dx, 03B4H               ; Adresse des Index-Registers
                mov     bl, 0                   ; Startwert für Regsterindex
                mov     cx, 12                  ; Zähler für 12 Parameter
L016:           mov     al, bl                  ; Index des Registers laden
                out     dx, al                  ; Indexregister programmieren
                inc     dx                      ; Adresse des Data-Registers jetzt in DX
                lodsb                           ; Wert für Register laden
                out     dx, al                  ; Register programmieren
                dec     dx                      ; Adresse des Index-Registers jetzt in DX
                inc     bl                      ; Index des nächsten Registers
                loop    L016                    ; nächstes Register programmieren
                mov     dx, 03B8H               ; Adresse des Display Mode Ctr. Ports
                mov     al, 28H                 ; Bildschirm wieder einschalten
                out     dx, al                  ; Display Mode Control Port setzen
                                                ;
                                                ; Text-Bildschirm löschen
                                                ;
                mov     ax, Base                ; Segment der Grafikkarte
                mov     es, ax                  ; Segment ins ES-Register kopieren
                mov     di, 0                   ; Startoffset des Textspeichers
                mov     cx, 2000                ; Zahl der zu schreibenden Worte
                mov     ax, 0                   ; Wert für Textspeicher
                rep     stosw                   ; Textspeicher löschen
                                                ;
                mov     sp, bp                  ; BP-Register nach SP kopieren
                pop     ds                      ; DS-Register restaurieren
                pop     bp                      ; BP-Register restaurieren
                ret                             ; Ende der Prozedur
                                                ;
Text_Data       db      61H, 50H, 52H, 0FH, 19H, 06H ; Registerwerte für Textmodus
                db      19H, 19H, 02H, 0DH, 0BH, 0CH ;
Leave_Graphic   endp                            ;
                                                ;
cseg            ends
                end
```

Bild 2. Die Routinen zum Wechseln des Video-Modus

```
Uses CRT;

Function HGC_Mode : Boolean; External;
Procedure Enter_Graphic; External;
Procedure Leave_Graphic; External;

{$L HGC_Mode}          { Object-Datei der Funktion HGC_Mode }
{$L Switch}            { Object-Datei der Funktionen Enter- und Leave_Graphic }

var Mode1, Mode2 : Boolean;

Begin
  Enter_Graphic;       { Aktivierung des Grafikmodus  }
  Delay (100);         { Warten bis Karte bereit ist  }
  Mode1 := HGC_Mode;
  Leave_Graphic;       { Aktivierung des Textmodus    }
  Delay (100);         { Warten bis Karte bereit ist  }
  Mode2 := HGC_Mode;
  Writeln (Mode1);
  Writeln (Mode2);
End.
```

Bild 3. Das Pascal-Demo-Programm zum Test der Funktion HGC_Mode

Wie kann die Lichtgriffel-Steuerung nun zur Lösung unseres Problems beitragen? Erstens läßt sich der Lichtgriffel rein softwaremäßig emulieren. Durch einen Zugriff auf den Port 03BBH wird das Lichtgriffel-Flipflop zurückgesetzt und durch einen Zugriff auf den Port 03B9H gesetzt. Abwechselndes Rücksetzen und Setzen des Flipflops simuliert die Impulse eines Lichtgriffels. Zum zweiten unterscheidet sich das Rahmen-Timing von Text- und Grafikmodus in einem wichtigen Punkt. Im Textmodus besteht jedes Textzeichen aus einer 9 Pixel breiten und 14 Pixel hohen Matrix. Bei der Darstellung von 80 Textspalten und 25 Textzeilen ergibt sich eine Auflösung von 720 × 350 Pixeln. Im Grafikmodus liegt dem Timing der Karte ein 16 Pixel breites und 4 Pixel hohes Textzeichen zugrunde. Bei einer Auflösung von 720 × 348 Pixeln unterteilt sich der Bildschirm rein rechnerisch in 45 Textspalten und 87 Textzeilen. Im Textmodus werden also 2000 (80 × 25) Textzeichen ausgegeben, während es im Grafikmodus 3915 (45 × 87) Textzeichen sind. Beim Auslösen des Lichtgriffel-Impulses in der rechten unteren Ekke des Bildschirms müssen die Register 16 und 17 des 6845 je nach Anzeigemodus eine Lichtgriffel-Position (14-Bit-Zahl mit Low-Byte in Register 17) Werte um 2000 bzw. 3915 enthalten.

Eine Applikation der Firma Hercules schlägt vor, den Lichtgriffel zu triggern, sobald das Bit 7 des Display-Status-Ports (E/A-Adresse 03BAH) den Beginn der vertikalen Rücklaufphase (VSync) anzeigt. Leider funktioniert dieses Verfahren bei einigen Hercules-kompatiblen-Karten nicht. Daher wurde bei dem im *Bild 1* gezeigten Assembler-Programm ein anderer Weg eingeschlagen. Hier wird in einer Schleife von mindestens 20 ms Dauer (50 Bilder pro Sekunde) ständig der Lichtgriffel-Impuls simuliert und die größte erhaltene Textposition ermittelt. Am Ende des Programms wird der Wert mit einem Grenzwert verglichen und damit die Unterscheidung von Text- und Grafikmodus getroffen. Das gezeigte Assembler-Programm ist für einen Aufruf durch Turbo-Pascal (*Bild 2* u. *Bild 3*) ausgelegt, läßt sich aber leicht auf eine andere Programmiersprache anpassen. □

PLD-Compiler vereinfacht Entwicklung

Um die Entwicklung hochkomplexer PLDs zu vereinfachen, muß ein brauchbares CAE-System einige wesentliche Dinge können: Es muß z.B. ein hierarchisches Design unterstützen, bei dem umfangreiche Schaltungen in einzelne Blöcke aufgelöst werden, damit der Entwickler kleinere, überschaubare Einheiten wie gewohnt bearbeiten kann. Diese Blöcke sollten dann verbunden oder in übergeordnete Blöcke integriert werden können. Außerdem sollten logisches und physikalisches Design vollständig voneinander getrennt sein.

Da der Funktionsumfang der Schaltungen ständig steigt, darf man von einem modernen CAE-Werkzeug außerdem die Fähigkeit zur Logiksynthese erwarten. Es darf nicht nur die Schaltplaneingabe verstehen, sondern auch die funktionale Aufgabenbeschreibung einer zukünftigen Schaltung.

Die Vorteile moderner CAE-Werkzeuge sind kürzere Entwicklungszeiten und änderungsfreundliche Schaltungsentwürfe. Außerdem lassen sich oft mehrere alte Entwürfe in einem neuen, komplexeren PLD zusammenfassen.

Aus dem Technologie-Zentrum Karlsruhe kommt jetzt die 3. Generation des CAE-Systems LOG/iC, mit dem die ASIC-Entwicklung trotz steigender Komplexität überschaubar bleibt. Das ganze Programmsystem ist kompatibel zu den bisherigen Produkten und je nach Bedarf zu konfigurieren.

LOG/iC besteht aus mehreren Compilern und eignet sich zum blockorientierten, strukturierten Design. Die einzelnen Blökke kann der Entwickler als getrennte Datensätze behandeln und Booleschen Gleichungen, Funktionstabellen, mit der FSM-Syntax oder mit Schaltplaneingabe beschreiben. Verschiedene gebräuchliche Stromlaufplan-Editoren arbeiten mit dem Programmsystem zusammen. Boolesche Gleichungen und Funktionstabellen können innerhalb eines Datensatzes gemischt werden.

Unmittelbar nach der Eingabe der Datensätze für die einzelnen Blöcke gibt die Konsistenzprüfung dem Entwickler Auskunft über logische Widersprüche in der Definition der zukünftigen Schaltung. „Denkfehler" werden so in einer frühen Phase des Entwicklungsvorgangs abgefangen. Hat der Linker die einzelnen Blöcke zu einer kompletten Schaltung verbunden, kann man die Schaltung bereits simulieren, obwohl sie bis dahin nur als logischer Entwurf existiert. Die Schaltung braucht dazu noch nicht auf einen bestimmten IC-Typ abgestimmt zu sein.

Bei der Auswahl der richtigen PLD-Bausteine wird der Entwickler durch ein Datenbank unterstützt. Die Datenbank enthält unter anderem ausführliche strukturelle und physikalische Daten von über 230 PLD-Bausteinen von mehr als 20 Herstellern.

Da beim Programmsystem LOG/iC der physikalische Entwurf vom logischen getrennt ist, kann der Anwender mit verschiedenen Compilern erstellte Datensätze für alternative Realisierungen wiederverwenden. Die Entwurfsmethodik bleibt weitgehend gleich, da die Entscheidung für eine bestimmte Realisierung erst sehr spät getroffen werden muß. Dadurch ist LOG/iC nicht nur ein wichtiges Werkzeug zur PLD-Entwicklung, sondern durch den Gate-Array-Compiler auch ein ideales Entwurfs- und Prototypen-System für Gate-Array-Design.

Das LOG/iC-System läuft auf IBM-kompatiblen PC und AT, verschiedenen Workstations und auf VAX-Computern. Über Schnittstellen ist das System leicht in eine bestehende CAE-Umgebung zu integrieren.

Christoph Engel, Andreas Grimm

Midi-Hardware für Amiga

In mc wurden schon mehrere Midi-Schaltungen vorgestellt. Die Schnittstelle für Commodores Amiga ermöglicht trotz ihres einfachen Aufbaus die Wahl aus einem großen Softwareangebot – und eröffnet so den Weg zu professionellen Anwendungen mit Synthesizern, Drum- und Keyboardcomputern.

Umgekehrt kann der Amiga natürlich auch als Speichermedium im Midi-Verbund dienen oder seine eigenen musikalischen Fähigkeiten unter Beweis stellen. Als Sound-Sampler schließlich eröffnet der Rechner eine Welt, die sonst nur mit weit höherem finanziellen Aufwand erreichbar ist. Dabei braucht man zum Nachbau der hier vorgestellten Schaltung nur etwas Zeit, ein wenig Geduld und kaum praktische Vorkenntnisse in Elektronik. Die Schaltung kann mit allen Amiga-Modellen (500, 1000 und 2000) eingesetzt werden. Und auch Probleme mit der Bauteile-Beschaffung dürfte es kaum geben.

Midi und Amiga

Die Amiga-Modelle sind serienmäßig mit vier voneinander unabhängigen Tonkanälen ausgestattet, die – paarweise zusammengefaßt – auch Stereo-Ausgaben in vom Computer bisher kaum gekannter Qualität erlauben. Mit acht Bit Auflösung reichen diese Tonkanäle allerdings für professionelle Ansprüche nicht aus: acht Bit entsprechen weniger als 50 dB Störspannungsabstand und erreichen damit nicht einmal HiFi-Qualität. Um Studio- oder Bühnenanforderungen gerecht zu werden, setzen Profis daher in steigendem Maße Erfassungs- und Ausgabegeräte mit Midi-Verbindung ein, an der ein Computer – wie etwa der Amiga – als Master fungiert. Der Computer kann als Mischer, Manipulator und Überwachungseinheit eingesetzt werden. Speziell der Amiga mit seinen Tonkanälen kann hier auch als Ausgabemedium dienen. So kann z. B. über Midi ein Keyboard angeschlossen und ein Synthesizer emuliert werden.

Jedes Gerät im Midi-Verbund besitzt drei Anschlüsse: In, Out und Thru. Nur die für den Masterbetrieb geeigneten Instrumente sind mit zwei Ausgangsbuchsen (Out) versehen, womit sich überlange Leitungen verhindern lassen. Die möglichen Konfigurationen eines Midi-Systems zeigt Bild 1: Konfiguration A ist die einfachste. Sie wird für den Midi-Omni-Mode verwendet, bei dem jedes angeschlossene Slave-Gerät (ohne Unterscheidung der heute üblichen Kanalnummern 1...16) reagiert.

Am gebräuchlichsten ist die Konfiguration B in Verbindung mit dem Poly- und Mono-Mode der Midi-Spezifikation. Im Poly-Mode erhält jedes angeschlossene Instrument eine eindeutige Kanalnummer, unter der es vom Bus-Master angesprochen wird. Theoretisch lassen sich so bis zu 16 Midi-Teilnehmer simultan verwalten.

Im Mono-Mode dagegen sind auch polyphone Geräte ansteuerbar, bei denen jede Stimme eine eigene Kanalnummer erhält. Dabei müssen die zu einem polyphonen Instrument gehörigen Kanalnummern in aufsteigender Reihenfolge festgelegt sein.

Diese Einschränkung gilt nicht mehr im Midi-Multi-Mode (C): Er vereint die Möglichkeiten des Poly- und Mono-Mode. Im Multi-Mode ist ein einzelner Kanal auch polyphon ansteuerbar; so müssen, um beispielsweise einen Dreiklang anzuspielen, nicht gleichzeitig drei Kanäle senden. Zu beachten ist allerdings, daß noch nicht alle Synthesizer den Multi-Mode-Betrieb beherrschen.

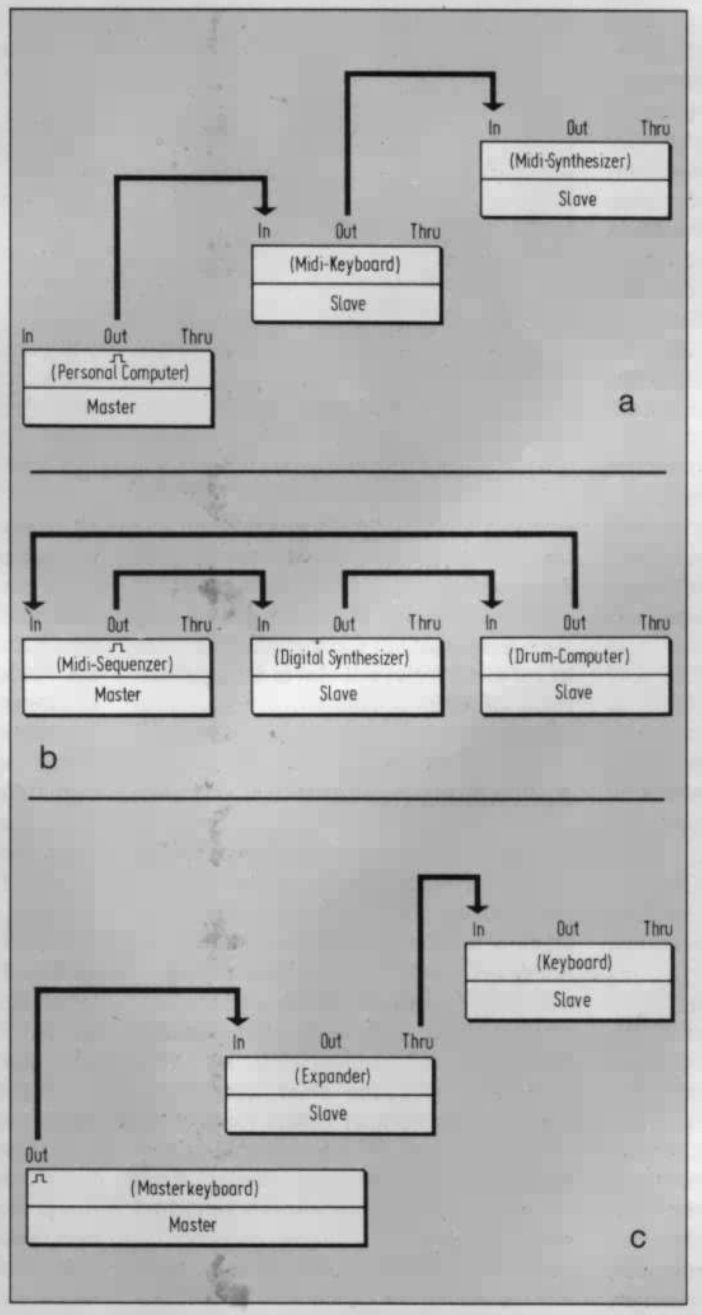


Bild 1. Drei mögliche Schaltungen des Midi-Busses: Konfiguration a für den Omni-Mode, Konfiguration b für Poly- und Mono-Mode sowie Konfiguration c für den modernen Multi-Mode

Elektrischer Aufbau

Elektrisch realisiert sind die Midi-Verbindungen mit Zweidraht-Leitungen, durch die ein Strom von 5 mA fließt. Der Ausgang (Out/Thru) eines Midi-Teilnehmers wird mit dem Ein-

gang (In) des nächsten Midi-Gerätes verbunden, wo ein Optokoppler für galvanisch getrennte Verbindungen sorgt. Der Ausgang des Optokopplers speist zum einen das Gerät selbst, zum anderen steht das Signal verstärkt an der Midi-Thru-Buchse dem nachfolgenden Midi-Teilnehmer zur Verfügung.

Das Zeitverhalten der Signalübertragung gleicht dem der RS232-C-Spezifikation, allerdings mit einer Baud-Rate von 31250 Bd. Da diese Übertragungsgeschwindigkeit in der RS232-C-Norm nicht vorgesehen ist, lassen sich viele handelsübliche Baudraten-Generatoren hier nicht einsetzen. Bei den Amiga-Rechnern ist dies jedoch ohne Änderung der Hardware möglich.

Die Datenübertragung erfolgt byteweise: Der Beginn wird durch ein Startbit angezeigt, gefolgt von den acht Datenbits und einem abschließenden Stopbit, das den Abstand zur nächsten Übertragung sichert. Etwa 3100 Bytes lassen sich so pro Sekunde übertragen, genug jedenfalls, um Musikinstrumente in Echtzeit anzusteuern.

Bei Midi-Verbindungen sollten die Leitungslängen das spezifizierte Maximum von 15 m nicht überschreiten. Da der Schnittstellenstrom nur 5 mA beträgt, nimmt die Störanfälligkeit bei Überschreiten dieses Maximums rapide zu. Die Kabel sollten abgeschirmt und mit einseitiger Masse (bei Midi-Out bzw. Midi-Thru) versehen sein. Besonders störanfällig ist auch der fünfpolige Diodenstecker, bei dessen Auswahl nur beste Qualität in Frage kommt. Das gilt auch für die Auswahl der Diodenbuchsen.

Schaltungsbeschreibung

Die Anschlüsse 4 und 5 der Midi-In-Buchse führen über zwei Widerstände an eine Diode und an den Optokoppler CNY-17, der durch die drei Bauteile vor Überspannung und Verpolung geschützt wird. Trotzdem ist es ratsam, den Optokoppler steckbar einzusetzen, da er Fehlanschlüsse schnell übel nimmt.

Das Eingangssignal gelangt über Pin 1 und 2 im CNY-17 an eine interne Leuchtdiode, die einen ebenfalls vergossenen Fototransistor ansteuert. Das Ausgangssignal wird nun für den Amiga-Anschluß mit einem Differenzverstärker invertiert sowie für den Midi-Thru-Anschluß nicht-invertiert verstärkt. Die Ansteuerung der beiden Midi-Out-Buchsen der Schaltung erfolgt über den TxD-Anschluß des Amiga. Das Signal wird von zwei Differenzverstärkern invertiert und über Schutzwiderstände an die Ausgangsbuchsen geleitet. Da der Differenzverstärker nur mit positiven Spannungen betrieben wird, werden die negativen Pegel des TxD-Anschlusses vorher abgeschnitten. Die gesamte Schaltung wird vom Amiga mit Strom versorgt, wobei auf die unterschiedliche Lage des 12-V-Anschlusses der Amiga-Modelle geachtet werden muß. Beim Amiga 1000 findet ein 25poliger Sub-D-Stecker Verwendung, die 12-V-Leitung liegt an Pin 23. Bei Amiga 500 und 2000 dagegen befindet sich die 12-V-Leitung an Pin 9; statt des Steckers ist hier eine 25polige Sub-D-Buchse erforderlich.

Alle Bauelemente auf einen Blick

1	Platine	Lochraster 9 x 5 cm
1	CNY-17	Optokoppler
1	LM324	
	oder TL84/74	
	oder LM2902	(vierfach OpAmp)
1	Sub-D-Stecker	25pol. für Amiga 1000 oder
1	Sub-D-Buchse	25pol. für Amiga 500/2000
4	Diodenbuchse	5pol. 180 Grad
2	1N4148	Si-Dioden
1	ZPD 5V1	Zenerdiode 5,1 V
8	220 Ω	1/8 W
2	2,2 kΩ	1/8 W
1	5,6 kΩ	1/8 W
1	10 kΩ	1/8 W
1	68 Ω	1/2 W
1	IC-Sockel	6pol.
1	IC-Sockel	14pol.
1	Kabel	4pol. zum Amiga

Obwohl die gesamte Schaltung mit einer Betriebsspannung von 5 V auskommt, ist auch beim Amiga 1000 der Einsatz des 12-V-Ausganges vorzuziehen. Der Rechner ist auf diese Weise mit einem Schutzwiderstand (68 Ω im Schaltbild) vor Kurzschlüssen geschützt.

Aufbauhinweise

Der Aufbau der Schaltung ist unkritisch: Er erfolgt auf einer kleinen Lochrasterplatte *(Tabelle)*, die über die 25polige Sub-D-Verbindung mit der seriellen Schnittstelle des Amiga gekoppelt wird. Bei Verwendung eines Platinensteckers müssen die Steckanschlüsse links und rechts der seriellen Schnittstelle frei bleiben; einfacher ist daher die Verwendung einer Kabelverbindung, die den Rechner zudem vor Zugbelastungen schützt. Da die Verbindung zum Amiga nur vier Leitungen erfordert, kann auf teure Quetschverbindungen verzichtet werden.

Obwohl die Schaltung absolut kurzschlußfest ist, sollten für Optokoppler und Differenzverstärker Sockel verwendet werden. Der 68-Ω-Widerstand muß ½ W Leistung vertragen, alle übrigen Widerstände kommen mit ⅛ W Belastbarkeit aus. Die beiden Schutzdioden der Schaltung können unter nahezu allen Siliziumtypen, etwa 1N4148, gewählt werden. Für den professionellen Einsatz ist zudem ein stabiles Gehäuse vorzusehen.

Nicht eindringlich genug kann hervorgehoben werden, daß für die 180°-Diodenbuchsen der Schnittstelle nur qualitativ hochwertige Steckverbindungen in Frage kommen. Außerdem ist eine stabile Befestigung der Buchsen am Gehäuse unabdingbar.

Fehlersuche

Funktioniert die Schaltung trotz sorgfältigsten Aufbaues nicht, empfiehlt sich folgendes Fehlersuchverfahren: Man trennt die Leitungen Pin 2 (TxD) und Pin 3 (RxD) des Amiga-Steckers von der Schaltung und prüft die 5-V-Spannung an der Zenerdiode und dem Differenzverstärker (Pin 4 bzw. 11). Als nächstes mißt man die Vergleichsspannung des Differenzverstärkers an Pin 2, 5, 10 und 12, die jeweils bei 1...2 V liegen sollte.

Falls alles in Ordnung ist, schließt man an die Pins 4 und 5 der Midi-Out-Buchsen eine Leuchtdiode an, wobei an Pin 5 der negative und Pin 4 der positive Pegel liegt. Empfehlenswert ist die Herstellung eines entsprechenden Diodensteckers mit Leuchtdiode, der zum Test der Midi-Schnittstelle auch in anderen Störfällen von großem Nutzen sein kann. Nun verbindet man die vom Amiga abgetrennte TxD-Leitung einmal mit dem 0-V-Anschluß der Zenerdiode (Leuchtdiode muß leuchten), dann mit dem 5-V-Anschluß an der Zenerdiode (Leuchtdiode darf nicht leuchten). Im Falle einer Fehlfunktion ist mit ziemlicher Sicherheit der Differenzverstärker defekt.

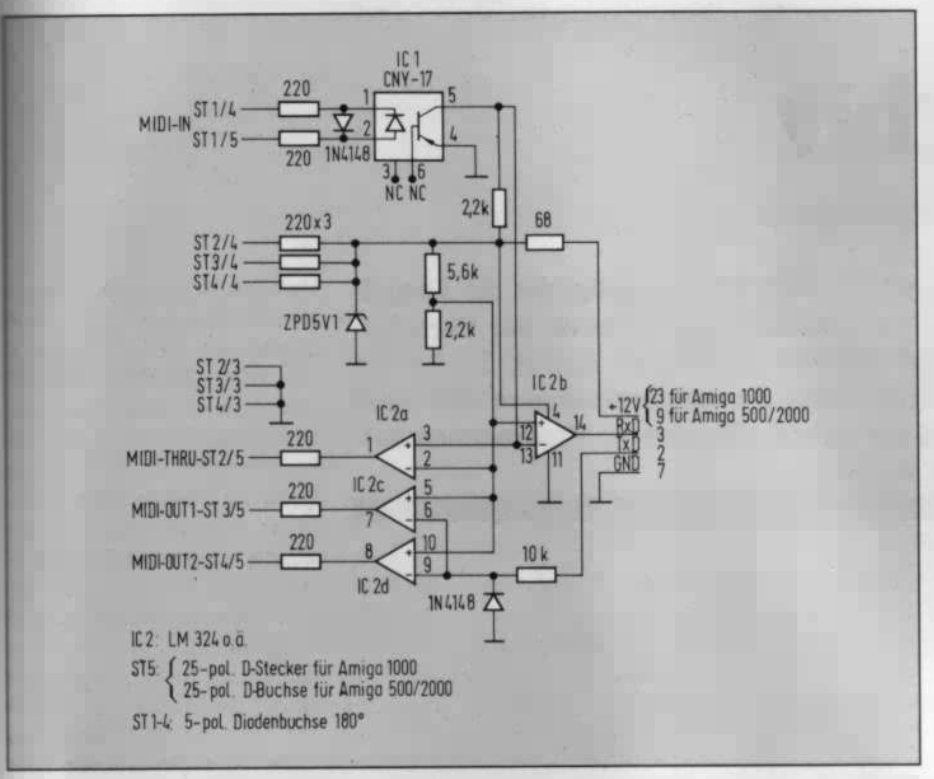


Bild 2. Die Midi-Hardware für Amiga-Rechner: Die einfache Schaltung eignet sich für die kommerzielle Amiga-Software und ist dabei sehr nachbausicher

Nun wiederholt man das Verfahren durch Einstecken des Teststeckers in die Midi-Thru-Buchse und verbindet einen der Midi-Out-Anschlüsse mit dem Midi-In-Anschluß (Loop-Schaltung). Auch hier muß das selbe Leuchtverhalten vorliegen, wenn die TxD-Leitung mit Masse bzw. 5 V verbunden wird. Falls jetzt alles der Erwartung entspricht, ist die Schaltung der Schnittstelle funktionstüchtig. Falls nicht, schließe man die Leuchtdiode an die vom Amiga abgetrennte RxD-Leitung bei gleicher Konfiguration (Loop-Schaltung) an, das Leuchtverhalten der Leuchtdiode muß nun umgekehrt sein. Falls nicht, ist der Optokoppler oder der Differenzverstärker defekt.

Midi-Software für den Amiga

Da jetzt die Schaltung sicherlich einwandfrei funktioniert, folgt nun eine kurze Beschreibung angebotener Software. Weit ab von jedem Anspruch auf Vollständigkeit werden im folgenden vier Programmpakete hervorgehoben, die ohne Probleme mit der beschriebenen Schnittstelle arbeiten. Ihr Preis liegt je nach Programm und Anbieter bei 140...350 DM.

Besonders komfortabel ist das „Pro Midi Studio" mit Soundsampler (Hersteller: Mimetics), das 16 verschiedene Midi-Geräte ansteuern und auf allen Midi-Kanälen gleichzeitig aufnehmen kann. Die Darstellung der Noten erfolgt in einem „Midi-Event", bei dem Dauer, Tonhöhe, Dynamik und vieles mehr in Form einer Tabelle eingegeben werden. Hervorzuheben ist der modulare Aufbau des Programms, mit dem Ein- und Ausgabemodule eröffnet und Musikdaten mit programmierbarem Klang an beliebige Midi-Kanäle weitergeleitet werden. Dieses Programm eignet sich besonders für den professionellen Tonbandschnitt.

Interessant an der Midi-Software „Sonix" (Hersteller: Aegis) ist die Amiga-Simulation eines Synthesizers, wobei sämtliche Parameter mit der Maus einstellbar sind. Die Speicherung der so erstellten Noten erfolgt in einem speziellen Format. Selbstverständlich können sie über Midi auch an andere Geräte ausgegeben werden.

Durch seine perfekte, vierfache Notendarstellung zeichnet sich die Software „Deluxe Music Construction Set" aus (Hersteller: Electronic Arts). Dabei sind alle möglichen Parameter wie Taktart, Notenschlüssel, Dynamik und Pausen einstellbar. Die Ein- und Ausgabe der Noten kann mit dem Amiga und dem Midi-Bus erfolgen – beispielsweise ist ein professionelles Keyboard mit Midi-Anschluß verwendbar. Die Speicherung der Notendaten erfolgt nach dem IFF-Standard, entwickelt von Commodore und Electronic Arts.

Midi-Marktlücke

Als abschließendes Beispiel sei hier das Programm „Dynamic Drums" erwähnt (Hersteller: New Wave), eine Drum-Software für den Amiga mit eingebautem Sequencer zur Erstellung, Abspeicherung und zum Abspielen von Arrangements. Dynamic Drums ist keine typische Midi-Software, weil sie auch eigenständig lauffähig ist. Wichtig für den Midi-Anwender ist hier die Synchronisationsfähigkeit des Programmes auf angeschlossene Midi-Partner, was ein automatisches Einstellen auf (auch veränderliche) Spielgeschwindigkeit ermöglicht. Erwähnenswert ist auch der äußerst günstige Preis dieses Programmes.

Obwohl das Angebot an Musik-Software für den Amiga nicht gerade gering ist, läßt es bei gehobenen Ansprüchen doch noch oft zu wünschen übrig – das betrifft auch die Midi-Software. Andererseits sind derzeit viele Programmierer mit dem Auffüllen dieser Marktlücke beschäftigt. Der Bedarf an brauchbarer Software ist sehr groß und die potentiellen Fähigkeiten des Amiga beinahe ebenso; jedenfalls sind sie bei weitem nicht erschöpft. Es wird also noch einiges in dieser Richtung zu erwarten sein. □

Jürgen Lipinski

Grafik-Archiv

Hercules-Grafiken auf Diskette gespeichert

Das Programm HERCCOPY ermöglicht die Abspeicherung von Grafikbildschirmseiten der Hercules-Karte auf Diskette und deren Aufruf. Es ist ein speicherresidentes Assemblerprogramm und kann während der Arbeit mit anderen Programmen aufgerufen werden.

Ein wesentlicher Unterschied zu vielen der bisher veröffentlichten speicherresidenten Programmen besteht darin, daß sowohl die komfortablen Funktionen des MSDOS-Interrupts 21H als auch Disk-I/O-Operationen, die nicht reentrant sind, genutzt werden können. Das wird durch die Prozeduren SAVE_SYSTEM und LOAD_SYSTEM erreicht, die Programmteile mit den problematischen System-Operationen einrahmen.

HERCCOPY (*Bild 1*) wurde mit dem MASM 1.25 entwickelt. Um das Programm durchsichtig zu gestalten, wurden alle Module des Programms als in sich abgeschlossene Prozeduren oder Makros definiert. Dies verringert wegen der damit verbundenen, häufigeren Stackoperationen in einigen Fällen die Arbeitsgeschwindigkeit, ermöglicht jedoch einen klar strukturierten Aufbau des Programms.

Die Routinen im Überblick

Die einzelnen Prozeduren und Macros sind, wie dem Listing zu entnehmen ist, zwar selbsterklärend, auf die Bedeutung der wichtigsten Module soll hier jedoch näher eingegangen werden.

Die Programmteile zur speicherresidenten Installierung des Programms und zur Umleitung des Tastaturinterrupts 16H wurden in der mc-Serie 'MSDOS intern' ausführlich erläutert. HERCCOPY verwendet die gleichen Routinen.

Die Prozeduren SAVE_SYSTEM und LOAD_SYSTEM rahmen das Anwenderprogramm ein und ermöglichen die Verwendung der Interrupt-21H-Funktionen. Die Funktionen dieses Softwareinterrupts legen temporäre Daten im DOS-Arbeitsspeicher ab der Adresse 00500H an. Unterbricht der Aufruf eines speicherresidenten Programms, das selbst Interrupt-21H-Funktionen verwendet, eine Funktion dieses Interrupts, so werden die Daten der unterbrochenen Funktion im DOS-Arbeitsspeicher überschrieben. Daraus ergibt sich die Notwendigkeit, die Daten des DOS-Arbeitsspeichers vor dem ersten Aufruf einer Interrupt-21H-Funktion im Anwenderprogramm in den Speicherbereich ab Offset SYSTEM_DATA temporär zu sichern (Prozedur SAVE_SYSTEM) und vor Verlassen des Anwenderprogramms den Ausgangszustand durch Rücktransfer der gesicherten Daten in den DOS-Arbeitsspeicher wiederherzustellen (Prozedur LOAD_SYSTEM).

Beide Grafikseiten können schnell gespeichert werden

Die Daten im Speicherbereich der Grafikseite 0 der HGC werden beim Aufruf der Menüs von HERCCOPY sowie beim Laden schon abgespeicherter Grafikseiten überschrieben. Um vor Verlassen von HERCCOPY die Ausgangsbildschirmseite restaurieren zu können (Prozedur GRAFIKSEITE0_LADEN), wird der Speicherbereich B0000H...B7FFFH in BUFFER1 temporär gesichert (Prozedur GRAFIK_SEITE0_SICHERN). Da BUFFER1 im Code-Segment von HERCCOPY liegt, besitzt das Programm den für ein Assemblerprogramm ungewöhnlich großen Speicherbedarf von 53 KByte.

Bei Eintritt in die Prozedur SICHERE_HERCULES_SCREEN befindet sich die Grafikseite 0 in BUFFER1, die Grafikseite 1, die bis dahin durch keine Daten überschrieben wurde, im Speicherbereich B8000H...BFFFFH. Beide Seiten können in verschiedene Diskdateien abgespeichert werden. An dieser Stelle ist zu beachten, daß einige Programme ihre Grafiken auf der Grafikseite 1 der HGC aufbauen. Um ein versehentliches Überschreiben schon abgespeicherter Bilder zu verhindern, prüft die Prozedur PROOF_DIRECTORY das Vorhandensein gleichnamiger Dateien. Die Prozedur OVERWRITE_ABSICHERUNG weist gegebenenfalls auf die Existenz dieser Datei hin.

LADE_HERCULES_SCREEN lädt ein schon abgespeichertes Bild von der Diskette oder Harddisk in BUFFER1, transferiert es von dort in den Speicherbereich der Grafikseite 0 der HGC (B0000H...B7FFFH) und aktiviert anschließend den Grafikmodus durch Aufruf der Prozedur GMOD. Kann die angegebene Datei auf dem Datenträger nicht gefunden werden, weist die Prozedur PROOF_DIRECTORY darauf hin. Der Weg über die Zwischenspeicherung des Bildes in BUFFER1 scheint umständlich zu sein, zumal der Buffer damit nicht mehr zur temporären Sicherung der Grafikseite 0 zur Verfügung steht und diese auf die Diskette zwischengespeichert wird (Prozeduren GPAGE0_TO_DISK und GPAGE0_FROM_DISK). Beim Austesten des Programms auf zwei XT-kompatiblen Rechnern (Commodore PC10 und Noname) sowie auf einem AT-kompatiblen Tandon zeigte sich, daß bei den langsameren XT-kompatiblen Rechnern der direkte Transfer der Bilddaten von der Diskette in den Bildschirmspeicher eine unvollständige Reproduktion des Bildes lieferte. Mit dem AT-kompatiblen Rechner traten diese Probleme nicht auf.

Die Prozedur MODUS_SETZEN gibt dem Anwender die Möglichkeit, wahlweise von HERCCOPY in den Textmodus (Bildschirmseite 0), den Hercules- Grafikmodus (Bildschirmseite 0) sowie den Hercules-Grafikmodus (Bildschirmseite 1) zurückzuwechseln. Diese Wahlmöglichkeit wurde geschaffen, da die Grafikprogramme, aus denen heraus HERCCOPY aufgerufen werden kann, in den verschiedensten Bildschirmmodi operieren.

Einfaches Starten des speicherresidenten Programms

Wie bei allen speicherresidenten Programmen erfolgt die Installierung von HERCCOPY von der MS-DOS-Betriebssystemsebene aus durch Eingabe von 'HERCCOPY'. Auch die Einbindung dieses Aufrufs in die AUTOEXEC.BAT Datei ist möglich. Eine wiederholte Installierung wird von HERCCOPY abgelehnt. Durch Drücken der Tastenkombination ALT-1 wird HERCCOPY aus den verschiedenen Programmen heraus gestartet. Mit PC-Paintbrush, Autocad, Lotus 1-2-3 und MS-Chart wurde es erfolgreich gete-

```
INDEX_REG               EQU  03B4H          ;REGISTER DES 6845
DATA_REG                EQU  03B5H
MODE_CONT               EQU  03B8H
CONFIGSWITCH            EQU  03BFH
TEXT                    EQU  20h            ;TEXTMODUS
BILDSCHIRM              EQU  10H            ;BILDSCHIRMINTERRUPT
CURRENT_RECORD          EQU  32
OFFSET_CURRENT_RECORD   EQU  32
OFFSET_FILE_SIZE        EQU  16
DOS                     EQU  21H
GPAGE1_OFFSET           EQU  8000H          ;GRAPHIKSEITE 1 DER HGC
GPAGE0_OFFSET           EQU  0000H          ;GRAPHIKSEITE 0 DER HGC
SEGMENT_GPAGE           EQU  0B000H         ;SEGMENT DER GRAPHIKSEITEN
READ_CHAR               EQU  01H            ;INT 21H-FUNKTION


CODE                    SEGMENT PARA PUBLIC 'CODE'
                        ORG  100H
ASSUME                  CS:CODE,DS:CODE,ES:CODE

START:                  JMP  INIT
;-----------------------------------------------------------------------
;                       M A C R O S
;-----------------------------------------------------------------------
BLANK_SCREEN            MACRO
                        PUSH AX
                        MOV  AH,00H
                        MOV  AL,3
                        INT  BILDSCHIRM
                        POP  AX
                        endm

GOTOXY                  MACRO X,Y
                        PUSH AX
                        PUSH DX
                        MOV  DH,Y
                        MOV  DL,X
                        MOV  AH,02H
                        INT  BILDSCHIRM
                        POP  DX
                        POP  AX
                        ENDM

Open                    macro
                        push ax
                        push dx
                        mov  dx,offset fcb
                        mov  ah,0fh
                        int  DOS
                        mov  file_exists,al
                        pop  dx
                        pop  ax
                        endm

close                   macro fcb
                        push ax
                        push dx
                        mov  dx,offset fcb
                        mov  ah,10H
                        int  DOS
                        pop  dx
                        pop  ax
                        endm

search_first            macro fcb
                        push dx
                        push ax
                        push bx
                        mov  ax,ds
                        mov  bx,offset buffer2      ;offset des ungeoeffneten fcb
                        set_dta ax,bx               ;fuer den fall, dass ein eintrag
                        mov  dx,offset fcb          ;gefunden wird
                        mov  ah,11h
                        int  DOS
                        mov  file_exists,al
                        pop  bx
                        POP  AX
                        pop  dx
                        endm

read_kbd                macro
                        push ax
                        push dx
                        mov  ah,08H
                        INT  DOS
                        pop  dx
                        pop  ax
                        endm

display                 macro string
                        push ax
                        push dx
                        mov  dx,offset string
                        mov  ah,09H
                        int  DOS
                        pop  dx
                        pop  ax
                        endm

get_string              macro limit,string
                        push ax
                        push dx
                        mov  string,limit
                        mov  dx,offset string
                        mov  ah,0AH
                        int  DOS
                        pop  dx
                        pop  ax
                        endm

create                  macro fcb
                        push ax
                        push dx
                        mov  dx,offset fcb
                        mov  ah,16H
                        int  DOS
                        pop  dx
                        pop  ax
                        endm

set_dta                 macro DTA_SEGMENT,DTA_OFFSET
                        push ax
                        push dx
                        push ds
                        MOV  AX,DTA_SEGMENT
                        mov  dx,DTA_OFFSET
                        MOV  DS,AX              ;DS = DTA_SEGMENT
                        mov  ah,1AH
                        int  DOS
                        pop  ds
                        pop  dx
                        pop  ax
                        endm
```

Bild 1. Assemblerlisting des Hercules-Hardcopy-Programms

```
set_relative_record     macro   fcb
                        push    ax
                        push    dx
                        mov     dx,offset fcb
                        mov     ah,24H
                        int     DOS
                        pop     dx
                        pop     ax
                        endm

ran_BLock_read          macro   fcb,count,rec_size
                        push    ax
                        push    dx
                        mov     dx,offset fcb
                        mov     cx,count
                        mov     ax,rec_size
                        mov     word ptr fcb[14],ax
                        mov     ah,27h
                        INT     DOS
                        pop     dx
                        pop     ax
                        endm

ran_block_write         macro   fcb,count,rec_size
                        push    ax
                        push    dx
                        mov     dx,offset fcb
                        mov     cx,count
                        mov     ax,rec_size
                        mov     word ptr fcb[14],ax
                        mov     ah,28H
                        int     DOS
                        pop     dx
                        pop     ax
                        endm

trans_filename          macro   filename,fcb
                        push    ax
                        push    dx
                        push    si
                        push    di
                        mov     si,offset filename
                        mov     di,offset fcb
                        push    es
                        push    ds
                        pop     es
                        mov     al,0h
                        mov     ah,29h
                        int     DOS
                        pop     es
                        pop     di
                        pop     si
                        pop     dx
                        pop     ax
                        endm

save_stack              macro
                        push    ax
                        push    bx
                        push    cx
                        push    dx
                        push    di
                        push    si
                        push    ds
                        push    es
                        endm

load_stack              macro
                        pop     es
                        pop     ds
                        pop     si
                        pop     di
                        pop     dx
                        pop     cx
                        pop     bx
                        pop     ax
                        endm
;=============================================================
;                e i n s p r u n g  ins hauptprogramm
;=============================================================
newint:                 or      ah,ah
                        jz      keyread
                        jmp     far [cs:oldint+2]

keyread:                pushf
                        call    far [cs:oldint+2]       ;tastaturabfrage
                        cmp     ax,7800h                ;alt-1? ja-->mainprogram
                        je      mainprogram             ;     nein-->return

return:                 iret

MAINPROGRAM:            cmp     al,cs:active
                        jne     return
                        PUSH    DS
                        PUSH    ES
                        MOV     AX,CS
                        MOV     DS,AX
                        MOV     ES,AX
                        MOV     OLDSS,SS        ;eigener stack
                        MOV     OLDSP,SP
                        MOV     SS,AX
                        MOV     SP,0FEH
                        PUSH    BP
                        PUSH    BX
                        PUSH    CX
                        PUSH    DX
                        PUSH    SI
                        PUSH    DI
                        sti
                        DEC     BYTE PTR ACTIVE
                        MOV     CS:ACTIVE,1
                        cli
                        CALL    HAUPTPROGRAMM
                        sti
LEAVE:                  MOV     CS:ACTIVE,0
                        POP     DI
                        POP     SI
                        POP     DX
                        POP     CX
                        POP     BX
                        POP     BP
                        MOV     SS,OLDSS
                        MOV     SP,OLDSP
                        POP     ES
                        POP     DS
                        mov     ah,0
                        JMP     keyread


HAUPTPROGRAMM           PROC
                        push bp
                        mov  bp,sp
                        CALL SAVE_SYSTEM        ;dos-arbeitsspeicher nach system_data
                        call program
                        CALL LOAD_SYSTEM        ;dos-arbeitsspeicher restaurieren
                        mov  sp,bp
                        pop  bp
                        RET
```

```
HAUPTPROGRAMM           ENDP

SAVE_SYSTEM             PROC
                        PUSH CX
                        PUSH DI
                        PUSH SI
                        PUSH ES
                        PUSH DS
                        PUSH DS
                        XOR  CX,CX
                        MOV  DS,CX                  ;DOS SEGMENT BEREICH
                        MOV  DI,OFFSET SYSTEM_DATA
                        MOV  SI,0500H               ;DOS DATA OFFSET
                        POP  ES                     ;ES=CS
                        mov  cx,3500h
                        CLD
                        REP  MOVSB
                        POP  DS
                        POP  ES
                        POP  SI
                        POP  DI
                        POP  CX
                        RET
SAVE_SYSTEM             ENDP

LOAD_SYSTEM             PROC
                        PUSH CX
                        PUSH DI
                        PUSH SI
                        PUSH ES
                        PUSH DS
                        MOV  SI,OFFSET SYSTEM_DATA
                        XOR  CX,CX
                        mov  ES,CX                  ;DOS SEGMENT BEREICH
                        MOV  DI,0500H               ;DOS DATA OFFSET
                        MOV  CX,3500H
                        CLD
                        REP  MOVSB
                        POP  DS
                        POP  ES
                        POP  SI
                        POP  DI
                        POP  CX
                        RET
LOAD_SYSTEM             ENDP

PROGRAM                 PROC
                        save_stack
                        call GRAFIKSEITE0_SICHERN   ;b0000-b7fff -->buffer1
                        CALL MAIN_MENU
                        CMP  AL,49                  ;bilder sichern ? ja-->saveh
                        JE   SAVEH
                        JMP  LOADH                  ;                 nein-->loadh
SAVEH:                  CALL SICHERE_HERCULES_SCREEN
                        JMP  LEAVE_PROGRAM
LOADH:                  CALL LADE_HERCULES_SCREEN

LEAVE_PROGRAM:          call MODUS_SETZEN           ;bildschirmmodus zum ausstieg setzen
                        load_stack
                        RET
PROGRAM                 ENDP

GRAFIKSEITE0_SICHERN    PROC
                        PUSH    CX
                        PUSH    ES
                        PUSH    DI
                        PUSH    SI
                        push    ds
                        pushf
                        push    ds
                        MOV     AX,0b000h           ;QUELLESEGMENT
                        MOV     DS,AX
                        pop     es                  ;zielsegment
                        MOV     SI,0000h            ;quelloffset
                        MOV     DI,offset buffer1   ;zieloffset
                        MOV     CX,32767            ;byteanzahl
                        CLD                         ;von links nach rechts uebertragen
                        REP     MOVSB
                        popf
                        pop     ds
                        POP     SI
                        POP     DI
                        POP     ES
                        POP     CX
                        RET
GRAFIKSEITE0_SICHERN    ENDP

GRAPHICPAGES_LADEN      PROC
                        PUSH    AX
                        PUSH    CX
                        PUSH    DS
                        PUSH    ES
                        PUSH    DI
                        PUSH    SI
                        pushf
                        MOV     AX,0b000h           ;zielSEGMENT
                        MOV     ES,ax
                        MOV     AX,0000h            ;ZIELoffset
                        MOV     di,ax
                        MOV     sI,offset buffer1   ;quelloffset
                        MOV     CX,32767            ;quellsegment=ds
                        CLD
                        REP     MOVSB
                        popf
                        POP     SI
                        POP     DI
                        POP     ES
                        POP     DS
                        POP     CX
                        POP     AX
                        RET
GRAPHICPAGES_LADEN      ENDP

MAIN_MENU               PROC

                        MOV     AH,00H
                        MOV     AL,7
                        INT     BILDSCHIRM
                        MOV     BH,00H
                        mov     x1,2
                        mov     x2,78
                        mov     y1,2
                        mov     y2,24
                        call    RAHMEN
                        GOTOXY  10,4
                        DISPLAY MAIN_MENU_ZEILE1
                        GOTOXY  10,10
                        DISPLAY MAIN_MENU_ZEILE2
                        GOTOXY  10,12
                        DISPLAY MAIN_MENU_ZEILE3
ZEICHEN_LESEN:          MOV     Bh,00H
                        GOTOXY  12,70
                        MOV     AH,read_char
                        INT     DOS
                        CMP     AL,49
                        JE      ENDE_MAIN_MENU
                        CMP     AL,50
                        JNE     ZEICHEN_LESEN
```

```
ENDE_MAIN_MENU:
                RET
MAIN_MENU       ENDP

GMOD            PROC
                save_stack

                XOR     AX,AX
                MOV     CX,13           ;bei 12 wird null nicht ausgeführt
                MOV     Dx,INDEX_REG
                MOV     si,OFFSET GTABLE
                add     si,cx           ;offset +13
                dec     si              ;offset +12
                dec     si              ;offset +11
GMODLOOP:       mov     ax,cx
                dec     ax
                dec     ax
                mov     dx,index_reg
                out     dx,al
                MOV     Dx,DATA_REG
                MOV     al,[SI]
                dec     si
                out     dx,al
                LOOP    GMODLOOP
                MOV     Dx,CONFIGSWITCH
                MOV     Al,3
                out     dx,al
                mov     dx,mode_cont
                MOV     AL,MODE         ;10 oder 138 (page 0 oder page 1)
                out     dx,al
                load_stack
                RET
GMOD            ENDP

tMOD            PROC
                save_stack
                XOR     AX,AX
                MOV     Dx,MODE_CONT
                MOV     AL,TEXT         ;20h
                out     dx,al
                MOV     CX,13
                MOV     Dx,INDEX_REG
                MOV     si,OFFSET tTABLE ;offset
                add     si,cx           ;offset +13
                dec     si              ;offset +12
                dec     si              ;offset +11
tMODLOOP:       mov     ax,cx
                dec     ax
                dec     ax
                mov     dx,index_reg    ;dx wird ja geändert
                out     dx,al
                MOV     Dx,DATA_REG
                MOV     al,[SI]
                dec     si
                out     dx,al
                LOOP    tMODLOOP
                MOV     Dx,CONFIGSWITCH
                MOV     Al,0
                out     dx,al
                mov     dx,mode_cont
                MOV     AL,28h
                out     dx,al
                load_stack
                RET
tMOD            ENDP
```

```
SICHERE_HERCULES_SCREEN PROC
                push    ax
                push    bx
                push    cx
                push    dx
                push    di
                push    si
                push    ds
                push    es
                CALL    MENU_SHS
                gotoxy  10,20
                display prompt1                 ;hgc-seite 0 sichern
                gotoxy  50,20
                get_string 15,filename
                trans_filename  filename[2],fcb
                CALL    OVERWRITE_ABSICHERUNG
                MOV     AX,OFFSET BUFFER1       ;SEITE 0 IST IM BUFFER
                SET_DTA DS,ax
                create  fcb
                mov     fcb[current_record],0
                set_relative_record fcb
                mov     ax,128
                mov     recordsize,ax
                mov     ax,256
                mov     recordnumber,ax
                ran_block_write fcb,recordnumber,recordsize
                close fcb
                gotoxy  10,22
                display prompt2                 ;hgc-seite 1 sichern
                gotoxy  50,22
                get_string 15,filename
                trans_filename  filename[2],fcb
                CALL    OVERWRITE_ABSICHERUNG
                mov     ax,segment_gpage
                SET_DTA AX,GPAGE1_OFFSET
                create  fcb
                mov     fcb[current_record],0
                set_relative_record fcb
                mov     ax,128
                mov     recordsize,ax
                mov     ax,256
                mov     recordnumber,ax
                ran_block_write fcb,recordnumber,recordsize
                close   fcb
                pop     es
                pop     ds
                pop     si
                pop     di
                pop     dx
                pop     cx
                pop     bx
                pop     ax
                RET
SICHERE_HERCULES_SCREEN ENDP

LADE_HERCULES_SCREEN    PROC
                push  ax
                push  bx
                push  cx
                push  dx
                push  di
                push  si
                push  ds
                push  es
                CALL  GPAGE0_TO_DISK
                CALL  MENU_LHS
```

```
FILENAME_INPUT:         gotoxy  10,22
                        display prompt3
                        gotoxy  45,22
                        get_string 15,filename
                        trans_filename      filename[2],fcb
                        CALL  PROOF_DIRECTORY                   ;existiert der file?
                        mov   bx,offset buffer1
                        mov   ax,cs
                        set_dta ax,bx                           ;SEGMENT_GPAGE,GPAGE0_OFFSET
                        MOV   AL,FILE_EXISTS
                        CMP   AL,0
                        JE    NO_FILE                           ;file nicht gefunden
                        JMP   CONTINUE_LH                       ;file gefunden

NO_FILE:                MOV   X1,40
                        MOV   X2,77
                        MOV   Y1,8
                        MOV   Y2,12
                        xor   bx,bx
                        CALL  RAHMEN
                        GOTOXY  42,9
                        DISPLAY FILE_NAME_NOT_FOUND
                        CALL  PRESS_KEY
                        MOV   CX,5
                        CALL  CLEAN_WINDOW
                        JMP   FILENAME_INPUT

CONTINUE_LH:            open  fcb
                        mov   fcb[current_record],0
                        set_relative_record fcb
                        mov   ax,128
                        mov   recordsize,ax
                        mov   ax,256
                        mov   recordnumber,ax
                        ran_block_read fcb,recordnumber,recordsize
                        close fcb
                        call  graphicpages_laden
                        MOV   MODE,10
                        call  gmod
                        read_kbd
                        CALL  GPAGE0_FROM_DISK
                        pop   es
                        pop   ds
                        pop   si
                        pop   di
                        pop   dx
                        pop   cx
                        pop   bx
                        pop   ax

HL_END:                 RET
LADE_HERCULES_SCREEN    ENDP

GPAGE0_TO_DISK          PROC
                        SAVE_STACK
                        mov   ax,cs
                        mov   bx,offset buffer1
                        set_dta ax,bx
                        create     fcb_TEMP
                        mov   fcb_TEMP[current_record],0
                        set_relative_record fcb_TEMP
                        mov   ax,128
                        mov   RECORDSIZE,ax
                        mov   ax,256
                        mov   recordnumber,ax
                        ran_block_write fcb_TEMP,RECORDNUMBER,RECORDSIZE
                        close fcb_TEMP
                        LOAD_STACK
                        RET
GPAGE0_TO_DISK          ENDP

GPAGE0_FROM_DISK        PROC
                        SAVE_STACK
                        mov   ax,cs
                        mov   bx,offset buffer1
                        set_dta ax,bx
                        trans_filename      filename_TEMP[2],fcb_TEMP
                        OPEN  fcb_TEMP
                        mov   fcb_TEMP[current_record],0
                        set_relative_record fcb_TEMP
                        mov   ax,128
                        mov   RECORDSIZE,ax
                        mov   ax,256
                        mov   recordnumber,ax
                        ran_block_READ fcb_TEMP,RECORDNUMBER,RECORDSIZE
                        close fcb_TEMP
                        LOAD_STACK
                        RET
GPAGE0_FROM_DISK        ENDP

MENU_SHS                PROC
                        PUSH  DX
                        PUSH  AX
                        push  bx
                        MOV   X1,5
                        MOV   X2,75
                        MOV   Y1,13
                        MOV   Y2,23
                        CALL  RAHMEN
                        mov   bh,00h
                        GOTOXY 10,15
                        DISPLAY MENU_SHS_ZEILE1
                        mov   bh,00h
                        GOTOXY 10,16
                        DISPLAY MENU_SHS_ZEILE2
                        mov   bh,00h
                        GOTOXY 10,18
                        DISPLAY MENU_SHS_ZEILE3
                        pop   bx
                        POP   AX
                        POP   DX
                        RET
MENU_SHS                ENDP

MENU_LHS                PROC
                        PUSH  DX
                        PUSH  AX
                        PUSH  BX
                        MOV   BH,00H
                        MOV   X1,5
                        MOV   X2,75
                        MOV   Y1,13
                        MOV   Y2,23
                        CALL  RAHMEN
                        GOTOXY 10,15
                        DISPLAY MENU_LHS_ZEILE1
                        GOTOXY 10,16
                        DISPLAY MENU_LHS_ZEILE2
                        GOTOXY 10,18
                        DISPLAY MENU_LHS_ZEILE3
                        POP   BX
                        POP   AX
                        POP   DX
                        RET
MENU_LHS                ENDP
```

```
OVERWRITE_ABSICHERUNG   PROC
                        SAVE_STACK
OVERWRITE_BEGIN:        CALL PROOF_DIRECTORY
                        MOV  AL,FILE_EXISTS
                        CMP  AL,1                ;FILE EXISTIERT
                        JE   OVERWRITE_?
                        JMP  ABSICHERUNG_END
OVERWRITE_?:            MOV     X1,40
                        MOV     X2,77
                        MOV     Y1,8
                        MOV     Y2,12
                        CALL    RAHMEN
                        GOTOXY  42,9
                        DISPLAY FILE_EXISTIERT
                        GOTOXY  42,10
                        DISPLAY OVERWRITE_FRAGE
YN:                     GOTOXY  75,11
                        MOV     AH,read_char
                        INT     DOS
                        CMP     AL,'N'
                        JE      NEW_FILENAME
                        CMP     AL,'n'
                        je      NEW_FILENAME
                        CMP     AL,'Y'
                        JE      OVERWRITE_EXCEPTED
                        CMP     AL,'y'
                        JE      OVERWRITE_EXCEPTED
                        JMP     YN
OVERWRITE_EXCEPTED:     XOR     BX,BX
                        MOV     CX,5
                        CALL CLEAN_WINDOW
                        JMP     ABSICHERUNG_END
NEW_FILENAME:           gotoxy  10,22
                        display prompt3
                        gotoxy  50,22
                        get_string 15,filename
                        trans_filename   filename[2],fcb
                        JMP     OVERWRITE_BEGIN
ABSICHERUNG_END:        LOAD_STACK
                        RET
OVERWRITE_ABSICHERUNG   ENDP

PROOF_DIRECTORY         PROC
                        PUSH    AX
                        PUSH    DX
                        PUSH    DS
                        PUSH    DI
                        PUSH    SI
PROOF_BEGIN:            search_first fcb
                        MOV     AL,FILE_EXISTS
                        CMP     AL,255           ;no file?
                        Je      FILE
                        MOV     FILE_EXISTS,1    ;file found =1
                        JMP     SEARCH_END
file:                   MOV     FILE_EXISTS,0    ;file NOT FOUND =0
SEARCH_END:
                        POP     SI
                        POP     DI
                        POP     DS
                        POP     DX
                        POP     AX
                        RET
PROOF_DIRECTORY         ENDP
```

```
CLEAN_WINDOW            PROC
FENSTER_LOESCHEN:       MOV     AL,7
                        ADD     AL,CL
                        xor     bx,bx
                        GOTOXY  40,aL
                        display leerzeile
                        LOOP    FENSTER_LOESCHEN
                        xor     ax,ax
                        xor     bx,bx
                        RET
CLEAN_WINDOW            ENDP

RAHMEN                  PROC
                        SAVE_STACK
                        XOR     BX,BX
                        XOR     CX,CX
                        MOV     CL,Y2
                        SUB     CL,Y1
                        MOV     AL,186
LOOP_SENKR:             MOV     BL,Y1
                        ADD     BX,CX
                        GOTOXY X1,BL
                        CALL    WRITE_CHAR
                        GOTOXY X2,BL
                        CALL    WRITE_CHAR
                        LOOP    LOOP_SENKR
                        MOV     AL,205
                        MOV     CL,X2
                        SUB     CL,X1
LOOP_WAAGER:            MOV     BL,X1
                        ADD     BX,CX
                        GOTOXY BL,Y1
                        CALL    WRITE_CHAR
                        GOTOXY BL,Y2
                        CALL    WRITE_CHAR
                        LOOP    LOOP_WAAGER
                        GOTOXY X1,Y1
                        MOV     AL,201
                        CALL    WRITE_CHAR
                        GOTOXY X2,Y1
                        MOV     AL,187
                        CALL    WRITE_CHAR
                        GOTOXY X1,Y2
                        MOV     AL,200
                        CALL    WRITE_CHAR
                        GOTOXY X2,Y2
                        MOV     AL,188
                        CALL    WRITE_CHAR
                        LOAD_STACK
                        RET
RAHMEN                  ENDP

MODUS_MENU              PROC
                        BLANK_SCREEN
                        MOV     X1,2
                        MOV     X2,79
                        MOV     Y1,2
                        MOV     Y2,24
                        CALL    RAHMEN
                        GOTOXY  10,4
                        DISPLAY MODUS1
                        GOTOXY  10,5
                        DISPLAY MODUS2
                        GOTOXY  5,10
```

```
                        DISPLAY MODUS3
                        GOTOXY  5,12
                        DISPLAY MODUS4
                        GOTOXY  5,14
                        DISPLAY MODUS5
                        GOTOXY  20,17
                        DISPLAY MODUS6
                        RET
MODUS_MENU              ENDP
MODUS_SETZEN            proc
                        SAVE_STACK
                        CALL    MODUS_MENU
                        GOTOXY  50,20
TASTE:                  MOV     AH,read_char
                        INT     DOS
                        CMP     AL,'1'
                        JE      TEXTMODUS
                        CMP     AL,'2'
                        JE      GRAFIKSEITE0
                        CMP     AL,'3'
                        JE      GRAFIKSEITE1
                        JMP     TASTE
TEXTMODUS:              CALL    GRAPHICPAGES_LADEN
                        CALL    TMOD
                        JMP     MODUS_END
GRAFIKSEITE0:           CALL    GRAPHICPAGES_LADEN
                        MOV     MODE,10
                        CALL    GMOD
                        JMP     MODUS_END
GRAFIKSEITE1:           CALL    GRAPHICPAGES_LADEN
                        MOV     MODE,138
                        CALL    GMOD


MODUS_END:              LOAD_STACK
                        RET
MODUS_SETZEN            ENDP

write_char              PROC                    ;CHARACTER WIRD IN AL ÜBERGEBEN
                        PUSH    AX
                        PUSH    BX
                        push    cx
                        xor     bx,bx           ;SCREENPAGE 0
                        mov     cx,1            ;CHARACTERANZAHL
                        mov     ah,10
                        int     bildschirm
                        pop     cx
                        POP     BX
                        POP     AX
                        RET
WRITE_CHAR              ENDP

PRESS_KEY               PrOC
                        GOTOXY  10,23
                        DISPLAY PRESS_MELDUNG
                        gotoxy  70,23
                        READ_KBD
                        RET
PRESS_KEY               ENDP

;=============================================================================
;                           v a r i a b l e n
;=============================================================================

PRESS_MELDUNG           DB   '*** press any key to continue ***$'
active                  db   0
FILENAME                DB   15h  DUP (' ')
FCB                     DB   37   DUP (?)
FILENAME_TEMP           DB   'BILD.TMP'
FCB_TEMP                DB   0,'bild    tmp'
                        db   28   DUP (?)

RECORDSIZE              DW           ?
OLDINT                  Dd   ?
oldpos                  dw   ?
oldsp                   dw   ?
oldss                   DW   ?
prompt1                 DB   'Filename for page 1 ?  $'
prompt2                 db   'Filename for page 2 ?  $'
PROMPT3                 DB   'Filename ?$'
MAIN_MENU_ZEILE1        DB   'MAIN MENU$'
MAIN_MENU_ZEILE2        DB   '1 = Save Hercules graphic$'
MAIN_MENU_ZEILE3        DB   '2 = Load Hercules graphic$'
MENU_SHS_ZEILE1         DB   'Saving Hercules Graphic Screen$'
MENU_SHS_ZEILE2         DB   '-----------------------------$'
MENU_SHS_ZEILE3         DB   'Output file                                $'
MENU_LHS_ZEILE1         DB   'Loading Hercules Graphic Screen            $'
MENU_LHS_ZEILE2         DB   '-----------------------------              $'
MENU_LHS_ZEILE3         dB   'Input file  ?                              $'
FILE_NAME_NOT_FOUND     dB   'Cannot find the file $'
FILE_EXISTIERT          DB   'There exists a file  $'
OVERWRITE_FRAGE         DB   'overwrite ? (Y/N)    $'
LEERZEILE               DB   '                                           $'
MODUS1                  DB   'S E L E C T I O N  O F  T H E  S C R E E N  M O D E$'
MODUS2                  DB   '-------------------------------------------------$'
MODUS3                  DB   '1 = TEXT mode$'
MODUS4                  DB   '2 = GRAPHIC mode (Page 0)$'
MODUS5                  DB   '3 = GRAPHIC mode (Page 1)$'
MODUS6                  DB   'select a mode!$'
recordnumber            dw   ?
X1                      DB   ?
X2                      DB   ?
Y1                      DB   ?
Y2                      DB   ?
MODE                    DB   2 dup (?)
FILE_EXISTS             DB   0
GTABLE                  DB   35H,2DH,2EH,07H
                        DB   5BH,02H,57H,57H
                        DB   02H,03H,00H,00H
ttable                  db   61h,50h,52h,0fh
                        db   19h,06h,19h,19h
                        db   02h,0dh,0bh,0ch
SYSTEM_DATA             DB   4000H DUP(?),'$'
bufferI                 db   7FFFH dup (?),'$'
buffer2                 db   37    dup (?)
;=============================================================================
;                           i n i t i a l i s i e r u n g
;=============================================================================

TEXTINI                 DB   13,10,'HERCCOPY    geschrieben v. Jürgen Lipinski'
                        DB   13,10,'AUFRUF:ALT-1'
                        DB   13,10,'$'
NOMORE                  DB   13,10,'Herccopy ist '
                        DB   'bereits installiert$'
                        DB   13,10,'$'


init:                   mov  ax,3516h
                        INT  DOS
```

```
                                mov  word ptr oldint,bx
                                mov  word ptr oldint[2],es
                                mov  si,offset newint
                                mov  di,bx
                                mov  cx,100
        repe                    cmpsb
                                je   only1
                                mov  dx,offset newint
                                mov  ax,2516h
                                INT  DOS
                                mov  dx,offset textINI
                                mov  ah,9
                                INT  DOS

                                MOV  DX,offset buffer1
                                add  dx,8500h
                                int  27h
        only1:                  mov  dx,offset nomore
                                mov  ah,9
                                INT  DOS
                                mov  ax,4c00h
                                INT  DOS


        code                    ends
                                end  start
```

stet. Das nach dem Start erscheinende Menü erlaubt das Laden bzw. Sichern von Hercules-Grafiken. Nach dem Sichern von Hercules-Grafiken kehrt das Programm automatisch zum Ausgangszustand zurück; nach dem Laden einer Grafik erscheint diese auf dem Bildschirm und erst nach Drükken einer beliebigen Taste wird der Ausgangszustand regeneriert.

Was tun mit den Bildern

Auf die mit HERCCOPY erzeugten Bilddateien können Turbo-Pascal-Programme direkt zugreifen. Das Listing des Beispielprogramms BILDLADER (*Bild 2*) ermöglicht den direkten Transfer der Bilddateien in den Speicher der Grafikseite 0 der HGC. Durch Erweiterung des Programms kann man eine „Diashow" mit vorbereiteten Bildern programmieren. □

```
PROGRAM BILDLADER;

const
  h_xmax      =  719;       { maximum nr. of steps in x }
  h_ymax      =  348;       { maximum nr. of steps in y }
  h_graph_base = $B000;     { graphic memory base address }
  h_graph_size = 32767;     { size of graphic memory }
  h_errptr : Integer = 0;   { Adresse Fehlerroutine }

type
  h_screen         = Array [0..h_graph_size] of Byte; { Bildschirm }
  h_file           = String[64];                      { Dateiname }

var
  Bildname  : h_file;
  ReadError : Integer;


procedure clearscreen;
{ Bildschirm loeschen }
var
  Screen : h_screen absolute h_graph_base:$0000;
begin
Fillchar(Screen,h_graph_size,0);
end;

procedure h_error(n, m : Integer); forward;

procedure hires;
{ Grafik einschalten }
const
  IndexReg = $3B4;  { 6845 Indexregister }
  DataReg  = $3B5;  { 6845 Datenregister }
  ModeCont = $3B8;  { 6845 mode Control Port }
  ConfigSw = $3BF;  { 6845 Configuration Switch }
  graphmode : Array [0..11] of Byte =
                  ($35,$2D,$2E,$07,$5B,$02,
                   $57,$57,$02,$03,$00,$00);
var
  regsel : Integer;
begin
  h_errptr := Errorptr;
  ErrorPtr := ofs(h_error);
  regsel := 0;
  Port[ModeCont] := 2;
  repeat
    Port[IndexReg] := regsel;
    Port[DataReg] := graphmode[regsel];
    regsel := succ(regsel);
  until regsel = 12;
  Port[ConfigSw] := 1;  { HALF Mode }
  delay(100);
  Port[ModeCont] := 10; { Einschalten }
  clearscreen;          { Schirm löschen }
end;

procedure textmode;
{ Grafik ausschalten }
const
  IndexReg = $3B4;  { 6845 Indexregister }
  DataReg  = $3B5;  { 6845 Datenregister }
  ModeCont = $3B8;  { 6845 mode Control Port }
  ConfigSw = $3BF;  { 6845 Configuration Switch }
  text_mode  : Array [0..11] of Byte =
                  ($61,$50,$52,$0F,$19,$06,
                   $19,$19,$02,$0D,$0B,$0C);
var
  regsel : Integer;
begin
  regsel := 0;
  Port[ModeCont] := $20;
  repeat
    Port[IndexReg] := regsel;
    Port[DataReg] := text_mode[regsel];
    regsel := succ(regsel);
  until regsel = 12;
  Port[ConfigSw] := 0;  { Diag Mode }
  delay(100);
  Port[ModeCont] := 40; { Einschalten }
  clrscr;               { Schirm löschen }
  ErrorPtr := h_errptr;
end;

procedure h_error;
{ bei runtime/io-Fehler in textmodus schalten }
begin
  textmode;
  Writeln('Fehler im Grafikmodus aufgetreten:');
  { nach Ende der Prozedur kommt die Turbo-Fehlermeldung }
end;

procedure loadpic(name : h_file; var error : Integer);
{ Laden Bild aus einer Datei }
var
  Buffer  : h_screen;
  Screen  : h_screen absolute h_graph_base:$0000;
  picfile : file;
begin
{$i-}
assign(picfile,name);
reset(picfile);
error := IOResult; if error <> 0 then exit;
blockread(picfile,Buffer,256);
error := IOResult; if error <> 0 then exit;
close(picfile);
error := IOResult;
{$i+}
if Error = 0 then Screen := Buffer;
end;

begin
  ClrScr;
  GotoXY(10,10);Write('NAME DES BILDES?');
  GotoXY(40,10);Read(Bildname);
  Hires;
  Loadpic(Bildname,ReadError);
  Readln;
  Textmode;
  if ReadError <> 0 then Writeln('Fehler: ',ReadError);
END.
```

Bild 2. So einfach kann man die Bilder von Pascal aus einlesen

Frank Jürgensen

Datei-Konverter für dBase

dBASE III Plus-Dateien aus beliebigen Random-Dateien erzeugt

dBase ist ein sehr leistungsfähiges und komfortables Datenbankverwaltungssystem, das eine flexible Auswertung und Sichtung von Datenbeständen ermöglicht. Mit einem Hilfsprogramm kann der Anwender mit dBase auf von anderen Programmen erzeugte Daten schnell zugreifen.

In der täglichen Arbeit mit dem PC kommt es häufig vor, daß Dateien von Programmen erzeugt werden. die nicht im gewünschten Maße der Auswertung zugänglich sind. Darunter gehören viele Programme zur Erfassung von Meßdaten, die häufig in Basic oder Pascal geschrieben, und die beim z. B. beim Kauf von A/D-Umsetzern mitgeliefert werden. Wenn auf diese Dateien von dBase oder einem kompatiblen Programm (Clipper) zugegriffen werden könnte, hätte man ein mächtiges Instrumentarium zur Auswertung der Datenbestände zur Hand. Genau diesem Zweck dient das Programm MAKEDB. Es stellt aus Random-Dateien, die von Turbo-Pascal oder verschiedenen Basic-Implementationen erzeugt wurden, fertige DBF-Dateien her. Entwickelt wurde es für den Turbo-Basic-Compiler und enthält auch einige anderweitig verwendbare Elemente – z. B. eine Prozedur zur Konvertierung von binären Turbo-Pascal-Zahlen in andere Formate und die boolesche Funktion FNFILEEXIST(Name), die prüft, ob eine Datei bereits existiert. Nützlich ist vielleicht auch der Trick, mit dem man von Turbo-Basic aus Dateien über 64 KByte Größe erzeugen kann. Der Vorgang der Umwandlung gestaltet sich recht einfach. MAKEDB liest die Steuerdatei PARAMAKE.DB und holt sich daraus die Informationen über die Struktur der zu konvertierenden Random-Datei und über die gewünschte dBase-Struktur.
In *Bild 1* ist eine Steuerdatei dargestellt, die einen Pascal-Record übersetzt.

Eine Steuerdatei erhöht den Bedienungskomfort

Die Steuerinformationen können mit einem Texteditor eingetragen werden. Die Steuerdatei darf Kommentarzeilen enthalten, die dem Anwender das Eintragen der Strukturdaten erleichtern – man muß sich nämlich strikt an die von MAKEDB verlangte Reihenfoge halten. Die Formatregeln sind aus der Beispielsteuerdatei PARAMAKE.DB (*Bild 2*) mit ihren ausführlichen Kommentaren leicht zu entnehmen. Wichtig ist natürlich, daß man die Satzstruktur der umzuwandelnden Datei kennt (Satzlänge, Reihenfolge und Typ der Felder). Bei Turbo-Pascal-Strings muß man berücksichtigen, daß in der erzeugten Datei jeder String um ein Byte länger ist, als in der Record-Deklaration des Programms. Bei der Definition der zu erzeugenden dBase-Struktur muß man sich an die Konventionen von dBase hinsichtlich der Feldnamen und -längen halten. Hierbei ist wichtig, daß das angegebene Format groß genug ist, ansonsten bekommt man ein etwas merkwürdiges Ergebnis. Da in dBase-Dateien numerische Werte in ASCII und ohne Exponentialformat existieren, sollte man vorsichtshalber sehr viele Nachkommastellen und noch viel mehr Stellen insgesamt angeben. Das Zahlenformat sieht wie bei der Write-Anweisung von Pascal aus. Andere Formate als „Zeichen" und „Numerisch" werden nach der dBase-Seite hin von MAKEDB nicht erzeugt. Weil man aber eine dBase-Datei erhält, kann man sämtliche gewünschten Formate (z. B. „Datum" oder „Logisch") später mit MODIFY STRUCTURE und REPLACE automatisch aus diesen Formaten erzeugen lassen.

1 MByte große dBase-Dateien schnell erzeugt

Das Programm (*Bild 3*) ist in Turbo-Basic geschrieben und hat in der abgedruckten Form schon dBase-Dateien mit knapp einem Megabyte Größe erzeugt. Es wird mit zwei Kommandozeilenparametern aufgerufen: Quelldatei und Zieldatei. Auf diese Weise läßt sich ein häufiges Konvertieren mit Batch-Dateien automatisieren. Der Variablen para$ sollte dann im Programm ein fester Dateiname für die Steuerdatei zuge-

```
Der RECORD sieht so aus:

TYPE satz = RECORD
                lfd :Integer;
                txt :String[20];
                num :Real;
            END;
VAR
akt : satz;
datei : FILE OF satz;

( ... )

Und die zugehörige Steuerdatei sieht so aus:

' Parameter für RNDDATEI.PAS (DBTEST)
' Msrealformat% : (Bei PASCAL-Typen irrelevant)
1
' Sourcesatzlaenge% :
29
' Sourcefelderzahl% :
3
' Quellformat
2,2
21,7
6,6
' Zielformat :
LFD,N,6,0
NAME,C,20,0
NULLWERT,N,19,9
```

Bild 1. So sieht eine Steuerdatei für einen Pascal-Record aus

wiesen werden (s. Kommentar im Listing). Das Programm dokumentiert sich weitgehend selbst. Es holt sich aus der Parameterdatei die Feldformate für die zu erzeugende dBase-Datei und erstellt für jedes Feld eine PRINT-USING-Maske (MASKES). Danach wird der Header der dBase-Datei zusammengestellt. Im Header steht auch eine Angabe über das Datum der letzten Aktualisierung. Diese Angabe hat innerhalb von dBase keine besondere Bedeutung – lediglich beim Aufruf von DISPLAY STRUCTURE wird dem Benutzer dieses Datum angezeigt. Da das Betriebssystem selbst darüber Buch führt, wann eine Datei erzeugt wurde, braucht man sich um die Richtigkeit dieses Header-Wertes nicht zu kümmern.

Andere Basic-Dialekte

Einige weitere Bytes des Headers sind reserviert für Einträge, die dBase später selbst vornimmt. Nachdem der Header geschrieben ist, übernimmt die Prozedur SCHREIBSATZ die Arbeit. Ihr steht in der Variablen PUFFERS das binäre Abbild eines Satzes der Quelldatei zur Verfügung. Entsprechend den aus der Parameterdatei gewonnenen Informationen über die Satzstruktur der Quelldatei wird PUFFERS zerpflückt und – je nach numerischem Format (SOURCECODE%) – decodiert. Die decodierten Daten werden dann unter Benutzung der oben erwähnten Maske in die dBase-Datei geschrieben. Das Programm läßt sich auch mit etwas Arbeit auf einen anderen Basic-Compiler-Dialekt übertragen. Es ist dabei jedoch auf das Format für Basic-Gleitkommazahlen zu achten. Da Turbo-Basic beide Formate (IEEE und Microsoft) lesen kann, gibt es ein in der Steuerdatei zu setzendes Flag: „microsoftrealformat%". Das muß dann eventuell konstant auf 1 gesetzt werden, und die cvms-Funktionsaufrufe werden durch cvs ersetzt. Wer es in Pascal umcodieren will, hat hier zumindest alles stehen, was das Programm tun muß.

Vorsicht bei Gleitkommazahlen

Mit den richtigen Format- und Typangaben in der Steuerdatei vermeidet man eigentlich alle Komplikationen, bis auf die „systembedingten": Gleitkommazahlen in einer Größenordnung von über 10^{18} verkraftet dBase nicht – und die Formatierungsmasken von MAKEDB ebenfalls nicht. Es entstehen dBase-Dateien, bei denen Daten in andere Felder überlappen. Wenn so etwas auftritt, liegt aber meistens nicht dieser Fall vor, sondern die dBase-Feldgröße wurde kleiner gewählt, als notwendig und möglich gewesen wäre. □

```
'        Parameterdatei für Programm  MAKEDB
'            Mit erläuternden Kommentarzeilen
'
' Alle Zeilen ohne - ' - am Anfang werden als Daten interpretiert!
'
'=============================================================
'   STRUKTURINFORMATIONEN ÜBER DIE ZU KONVERTIERENDE DATEI
'=============================================================
' So sähe die Random-Satz-Definition der Sourcedatei in Basic
' aus :
' open "R" , #1 ,"MESSDAT",64
' FIELD #1,2 AS F$(1),10 AS F$(2),10 AS F$(3),10 AS F$(4),_
' 4 AS F$(5),4 AS F$(6),4 AS F$(7),4 AS F$(8),4 AS F$(9)
'
' Die sich daraus ergebenden Daten werden im folgenden angegeben.
' Danach erfolgt die Festlegung des gewünschten DBASE-Datenstruktur
' Es können auch wahlweise TURBO-Pascal Files übertragen werden
' Erstes Steuerzeichen in dieser Datei (bedeutet bei Turbo-Dateien
' nichts, muß aber auftauchen) :
' Microsoftrealformat%  : Einfache und doppelte Real-Zahlen werden in
' Random-Dateien von Microsoft-Basics oft anders codiert als ieee :
' In diesem Fall ist hier 1 einzusetzen, ansonsten 0
'
1
'
' Sourcesatzlaenge% : Satzlänge der Random-Datei (kann länger sein als
' die Summe der Feldlängen)
'
64
'
' Sourcefelderzahl% : Wichtig für die Länge des dBase - Headers
' außerdem werden nur soviele der untenstehenden Satzparameter
' gelesen.
'
9
'
' Angaben zur Sourcesatzstruktur und zum Typ der Daten:
' die Sourcefeldlänge kann nur bei Sourcecode = 1 (Zeichen) und 7
(STRING)
' frei gewählt werden.Dabei ist zu beachten, daß bei Pascal-Strings
' die Länge um eins größer sein muß, als in der Record-Deklaration
' des Programms.
' ZEICHEN (Basic-String) Länge variabel, Code = 1
' INTEGER (Basic,TurboP) Länge = 2      , Code = 2
' LONG INTEGER (Basic)   Länge = 4      , Code = 3
' SINGLE (Basic)         Länge = 4      , Code = 4
' DOUBLE (Basic)         Länge = 8      , Code = 5
' REAL (TurboP)          Länge = 6      , Code = 6
' STRING[n] (TurboP)     Länge n+1      , Code = 7
'
' Die Reihenfolge des Auftretens in der Datei ist zu beachten.
' Einzuhaltendes Format:
' <Feldlänge>,<Feldcode>
'
2,2
10,1
10,1
10,1
4,4
4,4
4,4
4,4
4,4
'
'
'=============================================================
'   STRUKTURINFORMATIONEN ÜBER DIE ZU ERZEUGENDE DBASE-DATEI
'=============================================================
' Es müssen gleichviel Sätze spezifiziert werden wie oben.
' Einzuhaltendes Format:
'
' <Feldname>,<Feldtyp>,<Feldlänge>,<Nachkommastellen>
'
' keine Leerzeichen erlaubt; GROSSCHRIFT vorgeschrieben!
' Die Strukturvorschriften von DB3+ sind zu beachten; die Typ-
' angabe erfolgt durch N für "Numerisch" oder C für "Zeichen",
' "Memo" wird nicht unterstützt, "Logisch" und "Datum" muß man
' sich unter dBase mit MODIFY STRUCTURE und REPLACE aus dem Zeichen-
' typ erstellen.
'
' Sehr wichtig: Die Felder dürfen keinesfalls zu klein gewählt werden
'
LFDNUMMER,N,5,0
DATUM,C,10,0
CKENNZ,C,10,0
TKENNZ,C,10,0
NULLWERT,N,19,9
CWERT,N,19,9
TWERT,N,19,9
FEHLER,N,19,9
DIFFERENZ,N,19,9
```

Bild 2. Ein Beispiel für eine Steuerdatei

```
'_____________________ MAKEDB _____________________
' Programm zur Erzeugung einer dBase Datenbankdatei
' aus einer Random-Datei, Version 2.0
' Frank Jürgensen, Bremen, 11-87
' Zur Parametrierung muß eine Datei in vorgeschriebenem Format
' vorliegen, die vom Programm automatisch interpretiert wird
'_____________________________________________________
' Prozeduren und Funktionen
'_____________________________________________________
def FNfileexist%(file$)
 local true,false
 true = -1 : false = 0
    on error goto weiter
    open "I",#10, file$
    FNfileexist% = true
    on error goto 0
    close # 10
    exit def
   weiter:
    FNfileexist% = false
    resume hier
    hier:
    on error goto 0
    close #10
end def
'_____________________________________________________
sub parameter.lesen(para$)
' liest aus der Datei "para" die Steuerparameter für makedb ein

  shared microsoftrealformat%, sourcesatzlaenge%, sourcefelderzahl%
  shared sourcefeldlaenge%(), sourcecode%()
  shared feldname$(), typ$(), laenge%(), dez%()

  local zeile$,i%,komma1%,komma2%,komma3%

  ' öffnen der Parameterdatei

  on error goto parameterfehler

  open "I", #3, para$

  ' Daten aus den Zeilen lesen, die nicht durch -'- als Kommentar-
  ' zeilen gekennzeichnet sind

  call lies.datenzeile(zeile$)
  microsoftrealformat%=val(zeile$)
  call lies.datenzeile(zeile$)
  sourcesatzlaenge%=val(zeile$)
  call lies.datenzeile(zeile$)
  sourcefelderzahl%=val(zeile$)

  if sourcesatzlaenge%=0 or sourcefelderzahl% =0 then
     error 99    ' dann liegt nämlich falsche
  end if         ' Parametrierung vor
```

Bild 3. Das Konvertierungsprogramm macht Dateien für dBase mundgerecht

```
for i%=1 to sourcefelderzahl%
   call lies.datenzeile(zeile$)
   komma1%=instr(zeile$,",")
   sourcefeldlaenge%(i%)=val(mid$(zeile$,1,komma1%-1))
   sourcecode%(i%)=val(mid$(zeile$,komma1%+1))
next
for i%=1 to sourcefelderzahl%
     call lies.datenzeile(zeile$)
     komma1%=instr(zeile$,",")
     komma2%=instr(komma1%+1,zeile$,",")
     komma3%=instr(komma2%+1,zeile$,",")

     if komma2%-komma1% <> 2 then error 99 'wäre unplausibel

     feldname$(i%)=ucase$(mid$(zeile$,1,komma1%-1))
     typ$(i%)=ucase$(mid$(zeile$,komma1%+1,1))
     laenge%(i%)=val(mid$(zeile$,komma2%+1,komma3%-komma2%-1))
     dez%(i%)=val(mid$(zeile$,komma3%+1))
  next

  close # 3
  on error goto 0
exit sub
  '_____________________
  ' Fehlerbehandlung
  '_____________________
  parameterfehler:
  print
  print "Fehler bei der Übernahme der Strukturdaten aus "para$
  print "überprüfen Sie das Format! Alle Zeilen werden als Daten"
  print "interpretiert, wenn in der ersten Spalte kein ' steht."
  print "Die Reihenfolge der Angaben ist strikt einzuhalten."
  end

end sub
'_____________________________________________________
sub lies.datenzeile(zeile$)
' gibt in zeile$ die nächste Zeile ohne Kommentarzeichen zurück

  line input #3, zeile$
  while mid$(zeile$,1,1)="'"
     line input #3, zeile$
  wend
end sub
'_____________________________________________________
 sub pasrealkonv(pasreal$,real#)
                 'wandelt den globalen String pasreal$ (aus Pascal-
File)
                 'in den entsprechenden REAL-Wert um
                 '(es können Rundungsfehler auftreten)
   'Zerpflücken
    strlen% = len(pasreal$)
    for i% = 1 to strlen%
      byte%(7-i%) = asc(mid$(pasreal$,i%,1))
    next

   'Auswertung
    if strlen%=6 then 'Turbo-Pascal Format
       v%=byte%(1) AND &H80 'Vorzeichenbit
       e%=byte%(6)          'Exponent
```

```
      if e%>0 then
        byte%(1)=byte%(1) - v%
        f#=0
        for i%=1 to 5
          f#=f#+byte%(6-i%)*exp2(8*(i%-1))
        next
        faktor# = exp2(e%-129)
        f#=1+f#*exp2(-39)
        w#=faktor#*f#

        if v%>0 then w#=-w#
     else
        w#=0
      end if
    else
      PRINT "Fehler :Kein Turbo-Pascal-Real"
    end if
    real#=w#
end sub
'_____________________________________________________________
sub schreibsatz(puffer$)
' Schreibt den Inhalt von Puffer$ entsprechend den Struktur-
' parametern umgewandelt in die dBase - Datei

  shared sourcefelderzahl%,sourcecode%()
  shared maske$(),microsoftrealformat%,sourcefeldlaenge%()

  pufferposition%=1
  for i%=1 to sourcefelderzahl%
    select case sourcecode%(i%)
           case 1
        schreib$=_
                 mid$(puffer$,pufferposition%,_
                      sourcefeldlaenge%(i%))
        print #2,using maske$(i%);schreib$;
      case 2
        schreib%=_
                 cvi(mid$(puffer$,pufferposition%,_
                          sourcefeldlaenge%(i%)))
        print #2,using maske$(i%);schreib%;
      case 3
        schreib&=_
                 cvl(mid$(puffer$,pufferposition%,_
                          sourcefeldlaenge%(i%)))
        print #2,using maske$(i%);schreib&;
      case 4
        if microsoftrealformat%=0 then
          schreib!=_
                   cvs(mid$(puffer$,pufferposition%,_
                            sourcefeldlaenge%(i%)))
        else
          schreib!=_
                   cvms(mid$(puffer$,pufferposition%,_
                            sourcefeldlaenge%(i%)))
        end if
        print #2,using maske$(i%);schreib!;
      case 5
        if microsoftrealformat%=0 then
          schreib#=_
                   cvd(mid$(puffer$,pufferposition%,_
                            sourcefeldlaenge%(i%)))
        else
          schreib#=_
                   cvmd(mid$(puffer$,pufferposition%,_
                            sourcefeldlaenge%(i%)))
        end if
        print #2,using maske$(i%);schreib#;
      case 6
        pasreal$=mid$(puffer$,pufferposition%,_
                            sourcefeldlaenge%(i%))
        CALL pasrealkonv(pasreal$,schreib#)
        print #2,using maske$(i%);schreib#;

      case 7
        schreib$=_
                 mid$(puffer$,pufferposition%+1,_
                 asc(mid$(puffer$,pufferposition%,1)))

        print #2,using maske$(i%);schreib$;

    end select
  pufferposition%=pufferposition%+sourcefeldlaenge%(i%)
  next i%
end sub
'_____________________________________________________________
'                        HAUPTPROGRAMM
'_____________________________________________________________
print
print "MAKEDB  V 2.0 von Frank Jürgensen"
print
'_____________________________________________________________
' Analyse der Aufrufparameter
'_____________________________________________________________
commando$ = lcase$(command$)
blanc=instr(2 , commando$," ")
dbfext=instr(blanc + 1 , commando$,".dbf")

if blanc=0 or dbfext=0 then
   print"aufruf: makedb [<lw>:<pfad>]quelle [<lw>:<pfad>]ziel.dbf"
      end
end if

quelle$=mid$(commando$,1,blanc)
ziel$  =mid$(commando$,blanc+1)
print "wandelt            "quelle$
print "um in :            "ziel$
if FNfileexist%(ziel$) then
   print "Zieldatei existiert schon - bitte umbenennen"
   end
end if

while not FNfileexist(quelle$)
   print "Bitte Taste drücken für Zugriff auf "quelle$
   errn%=errn%+1 :if errn% = 3 then end
   print
   print "<ESC> für Abbruch " : a$=input$(1)
   if asc(a$) = 27 then end
wend
errn% = 0

input "Strukturdaten in :",para$

while not FNfileexist(para$)
```

```
   errn%=errn%+1 :if errn% = 3 then end
   print para$ " nicht gefunden"
   input "Erneute Angabe oder <Return> :",para$
   if len(para$) = 0 then end
wend
'-------------------------------------------------------------------
' Einlesen der Strukturinformationen zum Lesen der Random Datei
' und zum Erstellen der dBase - Struktur
'-------------------------------------------------------------------
   dim sourcefeldlaenge%(128), sourcecode%(128), feldname$(128)
   dim typ$(128), laenge%(128), dez%(128)

call parameter.lesen(para$)
'-------------------------------------------------------------------
' Festlegung der Ausgabemasken
'-------------------------------------------------------------------
for i% = 1 to sourcefelderzahl%
  select case typ$(i%)
    case "C"
     if laenge%(i%)>1 then
       maske$(i%) ="\"+space$(laenge%(i%)-2)+"\"
     else
       maske$(i%) ="&"
     end if
    case "N"
     if dez%(i%)>0 then
       maske$(i%) = string$(laenge%(i%)-dez%(i%)-1,"#")+"."+_
                    string$(dez%(i%),"#")
     else
       maske$(i%) = string$(laenge%(i%),"#")
     end if
    case else
     print "Nur numerische oder String-felder werden erzeugt!"
     print "Alles andere ist auf DBASE - Ebene zu konvertieren"
     close : end
  end select
next i%

dbsatzlaenge% = 0
for i = 1 to sourcefelderzahl%
    feldname$(i)=feldname$(i)+string$(11-len(feldname$(i)),chr$(0))
    dbsatzlaenge% = dbsatzlaenge% + laenge%(i)
next i
incr dbsatzlaenge%  ' jeder Satz hat ein leeres Byte für Löschmarke
'-------------------------------------------------------------------
open "R" , #1 ,quelle$,sourcesatzlaenge%

FIELD #1,sourcesatzlaenge% as puffer$
satzzahl&=lof(1)/sourcesatzlaenge%
print satzzahl& "Sätze werden konvertiert"
'-------------------------------------------------------------------
' Initialisierungsvariablen für dBase-Header:
'-------------------------------------------------------------------
dbversion$ = chr$(&h03)     'bei Memofeldern &h83

jahr=87
monat = 11
tag = 13

aktdatum$  = chr$(jahr)+chr$(monat)+chr$(tag)
satzzahl$ = mkl$(satzzahl&)

headerlaenge$ = mki$(32% + sourcefelderzahl%*32% + 1)
dbsatzlaenge$ = mki$(dbsatzlaenge%)
reserv$ = string$(20,chr$(0))
abschluss$ = chr$(13)
dateiende$=chr$(26)

felddatadr$ = string$(4,chr$(0))   'Felddatenadresse im Speicher
                                   ' (Inhalt hier unwichtig)

resfmb$ = string$(2,chr$(0))+chr$(1)+string$(11,chr$(0))
          ' ebenfalls Sonderinformationen
'-------------------------------------------------------------------
' Header schreiben
'-------------------------------------------------------------------
open "O",#2, ziel$
print #2, dbversion$;aktdatum$;satzzahl$;headerlaenge$;
print #2, dbsatzlaenge$;reserv$;


for i% = 1 to sourcefelderzahl%
   print #2, feldname$(i%);typ$(i%);felddatadr$;
   print #2, chr$(laenge%(i%));chr$(dez%(i%));resfmb$;
next i%

print #2, abschluss$;
' daten schreiben
  print " -->";: spalte% = pos : zeile% = csrlin
  zaehler& = 0  : turnuslimit& = 60000 'Sonderbehandlung, weil TB in
                                       'einem Rutsch immer nur 64 KB
                                       'in eine Datei schreiben kann.
'-------------------------------------------------------------------
' Hauptschleife : Konvertierung aller Sätze
'-------------------------------------------------------------------
while loc(1) < satzzahl&

  get #1
  incr zaehler&

  print using"#### Sätze  #######
Bytes";loc(1);zaehler&*dbsatzlaenge%;
  locate zeile%,spalte%

  print #2," ";              ' platz für löschmarke
  call schreibsatz((puffer$))
  if zaehler& * dbsatzlaenge% > turnuslimit&  then
     close #2                          'Sobald 60000 voll sind,
     open "A", #2, ziel$               'wird die Datei neu eröffnet.
     turnuslimit& = turnuslimit& + 60000
  end if
wend
'-------------------------------------------------------------------
print#2,dateiende$;
print
print "           Fertig"
close #2
close #1
end


'-------------------------------------------------------------------
 Ende MAKEDB
```

Christian Philipps

Tastaturabfrage leichtgemacht

Einige nützliche Assembler-Routinen und C-Funktionen

Jeder ernsthafte C-Programmierer steht irgendwann vor dem Problem, die Tastatur des PC in geeigneterer Art und Weise abfragen zu müssen, als dies mit den Standard-Bibliotheksfunktionen möglich ist. mc stellt einige Routinen zur komfortablen Tastaturabfrage in Assembler und C vor.

Für eine Reihe von Anwendungen ist die Ermittlung des Tasten-Codes über den Interrupt 16H völlig ausreichend. Möchte man sich jedoch bei der Gestaltung der Benutzeroberfläche von selbstgeschriebenen Programmen an bereits weitverbreitete Vorgehensweisen anlehnen, erreicht man hier schnell die Grenzen.

Viele Programme reagieren bereits auf das Betätigen von Tasten, die im Normalfall keinen lesbaren Tastatur-Code erzeugen. Die Behandlung der Alt-Taste in der Programmierumgebung des Quick-C-Compilers ist ein gutes Beispiel. Befindet man sich im Editor und drückt nur die Alt-Taste, wird die Menüleiste aktiviert. Befindet man sich in einem Menü und betätigt ausschließlich diese Taste, schaltet Quick-C um eine Stufe in der Menühierarchie zurück.

Weitere Beispiele findet man ebenfalls bei der Bedienung von Quick-C. Soll z. B. das Wort rechts vom Cursor markiert werden, kann hierfür im Quick-C-Editor die Tastenkombination Ctrl-Shift-Pfeil-rechts verwendet werden. Die Funktion 0 des Interrupt 16H (Read character from keyboard) liefert für diese Tastenkombination jedoch denselben Wert wie für die Kombination „Ctrl-Pfeil-rechts". Letztere wiederum bewegt im Quick-C-Editor nur den Cursor um ein Wort nach rechts, jedoch ohne es zu markieren. Hier scheinen Unterscheidungen getroffen zu werden, die über das Auswerten der Scan-Code-Zeichen-Kombination, wie sie vom Interrrupt 16H geliefert wird, weit hinausgehen.

Im folgenden stelle ich eine Lösung vor, die nicht nur die gerade angesprochenen Probleme aus der Welt schafft, sondern auch weitgehend kompatibel zu der unter OS/2 verfügbaren Systemfunktion KbdCharIn ist, wenngleich deren gesamter Funktionsumfang unter DOS natürlich nicht abbildbar ist. Sie liefert neben der gewohnten Zeichen-Scan-Code-Kombination ein zusätzliches Wort zurück, in dem Bit-codiert der Zustand sämtlicher Shift- und Lock-Tasten abgelegt ist. Mit dieser Zusatzinformation ist die Tastenkombination „Ctrl-Shift-Pfeil-rechts" von der Kombination „Ctrl-Pfeil-rechts" zu unterscheiden. Ferner kann dieses Statuswort auch dann über die Funktion KbdCharIn() abgerufen werden, wenn kein Tastencode zur Bearbeitung bereitsteht. So ist auch das Drücken der Alt-Taste allein wahrnehmbar, ohne ins System eingreifen zu müssen.

Wie implementiert man eine solche Funktion unter DOS? Bedenkt man, daß Tasten wie z. B. die Alt-Taste nur auf der untersten Ebene der Tastaturbehandlung – dem Interrupt 9H – registrierbar sind, liegt der Gedanke nahe, eine neue Interrupt-Service-Routine zu entwerfen. Die in *Bild 1* dargestellte Assembler-Routine KBISR.ASM besteht hauptsächlich aus einer neuen Interrupt-Service-Routine für den Interrupt 9H.

Auf unterster Ebene

Leider ist die Behandlung der Tastatur auf der untersten Ebene alles andere als trivial, wovon man sich durch einen Blick in das ROM-BIOS-Listing sehr schnell überzeugen kann. Erschwerend kommt hinzu, daß die von einer PC/XT-Tastatur gelieferten Werte nicht mit denen von einer AT-Tastatur übereinstimmen und sich je nach Hersteller bei kompatiblen Rechnern weitere Unterschiede ergeben. Dafür, daß letztendlich in der Anwendung einheitliche Tasten-Codes das Licht der Welt erblicken, zeichnet im Normalfall das BIOS verantwortlich. Die für diese Aufgabe zuständige Interrupt-Service-Routine holt den Hardware-Code der Tastatur ab und wertet diesen anhand ROM-interner Tabellen aus. Wenn sich aus dieser Information eine brauchbare ASCII-Zeichen-Scan-Code-Kombination bilden läßt, wird diese in den BIOS-Tastatur-Puffer gestellt. Dort bleibt sie, bis sie über den Interrupt 16H abgerufen wird. Der Zustand der Shift- und Lock-Tasten (gedrückt/nicht gedrückt und ein/aus) wird nur im ROM-BIOS-Datenbereich protokolliert. Dort befindet sich an der Adresse 40:17H ein Statuswort, dessen Bits jeweils den Zustand einer Shift- bzw. Lock-Taste widerspiegelt. Dieses Statuswort spielt eine wichtige Rolle bei der Umsetzung der Hardware-Codes in ASCII-Zeichen und liefert uns im Hinblick auf die Problemstellung alle zusätzlichen Informationen.

Mit der Funktion 2 des Interrupt 16H (Return keyboard flags) kann man einen Teil – genau die Hälfte – der Shift-Flags ermitteln. Leider hat diese Vorgehensweise einen entscheidenden Nachteil: Die Aktionen „Tastencode abrufen" und „Shift-Status-abfragen" stehen in keinem unmittelbaren Zusammenhang. Weil das BIOS alle Tastendrücke in einem eigenen Puffer zwischenspeichert und diese dort im Prinzip stundenlang auf ihre Abholung warten können, ist nicht sichergestellt, daß der nach dem Anfordern von Scan-Code und Zeichen abgefragte Shift-Status genau den Zustand widerspiegelt, der beim Betätigen der jeweiligen Taste vorlag.

Jonglieren mit dem Interrupt 9H

An diesem Punkt setzt die erste Komponente meines Lösungsvorschlages an. Das Assembler-Modul KBISR.ASM enthält eine neue Interrupt-Service-Routine für den Tastatur-Interrupt (9H). Anstatt jedoch die mühsame Kleinarbeit der Code-Umsetzung selbst zu verrichten, simuliert diese einen Interrupt, der vom ursprünglichen Interrupt-Handler bedient wird. Dieser verrichtet wie gewohnt seinen Dienst, d. h. aktua-

```
;;;
```

```
                PUBLIC _InitKbdSys
_InitKbdSys     PROC FAR

                push    bp                      ; Standard-Intro
                mov     bp,sp
                push    ds                      ; veränderte Register
                push    es                      ; sichern
                mov     ax,[bp].stackofs        ; Adresse des privaten
                mov     my_sp,ax                ; Stacks wurde als Parameter
                mov     ax,[bp].stackseg        ; übergeben --> merken
                mov     my_ss,ax

                mov     ah,35H                  ; Hole alten IV
                mov     al,9
                int     21H
                mov     cs:isr9seg,es           ; und bewahre ihn gut auf
                mov     cs:isr9ofs,bx
                mov     al,9                    ; Installiere neue ISR
                mov     ah,25H
                push    cs
                pop     ds
                mov     dx,offset MyIsr9
                int     21H
                pop     es                      ; Segmentregister wieder-
                pop     ds                      ; herstellen
                pop     bp                      ; BP restaurieren
                ret

_InitKbdSys     ENDP

;=============================================================================

; EndKbdSys deaktiviert das Keyboard-Subsystem; die Original-ISR wird
; wieder eingehängt!.

                PUBLIC _EndKbdSys
_EndKbdSys      PROC FAR                        ; deinstalliere Keyboard-Handler

                push    bp                      ; Standard-Intro
                mov     bp,sp
                push    ds                      ; DS wird verändert
                mov     al,9                    ; Installiere Original-ISR
                mov     ah,25H
                mov     dx,cs:isr9ofs
                mov     ds,cs:isr9seg
                int     21H
                pop     ds                      ; DS wiederherstellen
                pop     bp                      ; BP wiederherstellen
                ret

_EndKbdSys      ENDP

;=============================================================================
; Private Interrupt-Service-Routine: Diese ISR wird bei jedem Tastendruck
; angesprungen.
; Von Natur aus faul, reicht sie den Tastendruck zunächst einmal zur Bear-
; beitung an die Original-ISR weiter und prüft dann hinterher über den
; Interrupt 16H ab, ob dieser Interupt zu einem brauchbaren Zeichen geführt
; hat. Nur wenn dies der Fall ist, wird der Tastencode eingesammelt und mit-
; samt aller Shift-Statusinformationen an den C-Queue-Handler weitergereicht.
; Um größtmögliche Übereinstimmung der Shift-Flags mit OS/2 zu erreichen und
; um auch die rechte Ctrl-Taste auswerten zukönnen (was bei nicht allen AT-
; Kompatiblen möglich ist), werden die Shift-Flags ggf. noch ein wenig
; manipuliert.
; Über ein Busy-Flag wird sichergestellt, daß die Routine nicht ein wei-
; teres Mal durchlaufen wird, wenn gerade die Bearbeitung eines Zeichens
; noch im Gange ist.

MyIsr9          PROC FAR

                cmp     cs:isr9busy,0           ; Bereits aktiv?
                jz      Isr91                   ; nein!
                JmpOldIsr cs:isr9ofs            ; ja, Interrupt durchreichen
Isr91:
                mov     cs:isr9busy,1           ; Busy-Flag setzen
                PushRegs                        ; Register sichern
                EmulInt cs:isr9ofs              ; und Original-ISR aufrufen
                mov     ax,DGROUP               ; Datensegment für C setzen
                mov     ds,ax
                mov     es,ax
                cli                             ; privaten Stack einrichten
                mov     ext_ss,ss
                mov     ext_sp,sp
                mov     ss,my_ss
                mov     sp,my_sp
                sti

; In der nachfolgenden Schleife werden alle im BIOS-Tastaturpuffer bereit-
; stehenden Zeichen abgeholt und an den Keyboard-Queue-Handler weiterge-
; reicht.
Isr92:
                mov     dx,es                   ; ES sichern
                mov     ax,BiosDataSeg          ; ES auf BIOS-Datenbereich
                mov     es,ax                   ; richten
                mov     bx,es:[BiosShiftStatus] ; Shift-Status -> BX
                mov     es,dx                   ; ES wiederherstellen

; Manipulation der Shift-Flags
                and     bx,NRightCtrlDown       ; SysReq ausmaskieren
                test    bx,CtrlShift            ; Control gedrückt?
                jz      Isr93                   ; nein!
                test    bx,LeftCtrlDown         ; ja - linke Ctrl-Taste ?
                jnz     Isr93                   ; ja, also weiter im Text
                or      bx,RightCtrlDown        ; nein, also Bit setzen,
                                                ; da das BIOS dies nicht tut
Isr93:
                mov     ah,1                    ; Frage ob Zeichen verfügbar
                int     16H
                jz      Isr94                   ; ZF=1 --> kein Zeichen
                mov     ah,0                    ; Lies Zeichen + Scancode
                int     16H                     ; Zeichen und Scan-Code --> AX

                push    bx                      ; Shift-Status auf Stack
                push    ax                      ; Zeichen+Scancode auf Stack
                call    _KbdQueHandler          ; Aufruf der C-Funktion
                add     sp,4                    ; Parameter entfernen
                jmp     Isr92                   ; Sprung zum Schleifenanfang
Isr94:
                cli                             ; alten Stack restaurieren
                mov     ss,ext_ss
                mov     sp,ext_sp
                sti
                mov     cs:isr9busy,0           ; Busyflag zurücksetzen
                PopRegs                         ; Register wiederherstellen
                iret                            ; zurück zum Ausgangspunkt
MyIsr9          ENDP

;================ Ende des Codesegmentes ======================================

KBISR_TEXT      ENDS
                END
```

lisiert den BIOS-Shift-Status oder plaziert eine Kombination aus ASCII-Zeichen und Scan-Code im BIOS-Tastaturpuffer. Anschließend kehrt er wieder zu der Instruktion innerhalb der neuen Interrupt-Service-Routine zurück, die auf den simulierten Interrupt folgt. An dieser Stelle kommt der Interrupt 16H zu seinem Recht: Wurde von der Original-Interrupt-Service-Routine ein Zeichencode im BIOS-Puffer abgelegt, wird dieser nun über den Interrupt 16H ausgelesen, durch den aktuellen Zustand des BIOS-Shift-Status (beide Bytes) ergänzt und als Doppelwort auf den Stack gelegt. Anschließend erfolgt der Aufruf der C-Funktion KbdQueHandler(), die diese vier Bytes als „Unsigned-long"-Parameter auswertet und in einem eigenen Ringpuffer abspeichert. Nach getaner Arbeit verabschiedet sich die neue Interrupt-Service-Routine und springt mit einer IRET-Instruktion zum unterbrochenen Programm zurück.

Ich möchte nicht versäumen darauf hinzuweisen, daß ich den Shift-Status des BIOS vor der Übergabe noch ein wenig „verdrehe". Zu dieser Maßnahme habe ich mich entschlossen, als ich feststellen mußte, daß das Phoenix-ROM-BIOS meines AT-kompatiblen das Bit für „rechte Ctrl-Taste gedrückt" niemals setzte. Ferner wurde ein zusätzliches Bit dauerhaft gesetzt, wenn ich Alt-SysAbfr drückte. Durch diese „Mogelei" konnte ich diese Klippen halbwegs umschiffen, womit jedoch keinesfalls gesagt ist, daß diese Maßnahmen für alle übrigen Rechner der AT-Klasse gleichermaßen ratsam ist. Da diese Stelle im Quellcode durch einen Kommentar gekennzeichnet ist, sollte es keine Probleme bereiten, die betreffenden Anweisungen zu streichen.

Neue Interrupt-Service-Routine

Betrachtet man sich den neuen Interrupt-Handler (Prozedur MyIsr9) genauer, fällt auf, daß sich dort noch ein wenig mehr abspielt. Zum einen wird für die C-Funktion KbdQueHandler() zunächst ein eigener Stack eingerichtet (zur Sicherheit) und zum anderen stellt ein Busy-Flag (IsrBusy) sicher, daß der Interrupt-Handler nicht durch einen weiteren Tastendruck erneut angesprungen wird, bevor er seine Arbeit ordnungsgemäß beendet hat. Letzteres ist zwingend erforderlich, da durch den simulierten Interrupt auf die BIOS-Interrupt-Service-Routine der Interrupt-Controller bereits wieder freigegeben wurde und somit durch den nächsten Tastendruck wiederum der Interrupt 9H ausgelöst wird. Da MyIsr9 nicht reentrant ist, würde der erneute Aufruf zu einem gewaltigen Chaos führen.

```
/*////////////////////////////////////////////////////////////////////////////
///       Allgemeinverwendbare Makro-Definitionen                            ///
///       Modul: SYSMAC.H                                                    ///
///       Autor: Christian Philipps  Stand: 4/88                             ///
////////////////////////////////////////////////////////////////////////////*/

#define TRUE            1
#define FALSE           0
#define BOOLEAN         int
#define NULL_CP         (char *)NULL
#define NULL_FP         (FILE *)NULL

/* Speicherplatzzuweisung mit automatischer Typkonvertierung */
#define MALLOC(TYP,SIZ) (TYP *)malloc((SIZ))

/* Speicherplatzzuweisung für ein Element eines bestimmten Typs */
#define MTALLOC(TYP) MALLOC(TYP,sizeof(TYP))

/* Speicherplatzzuweisung mit automatischer Typkonvertierung und Initialisierung
** auf 0H */
#define CALLOC(TYP,SIZ) (TYP *)calloc(1,(SIZ))

/* Speicherplatzzuweisung für ein Element eines bestimmten Typs
** Der Speicher wird mit Hexa 0 gefüllt. */
#define CTALLOC(TYP) CALLOC(TYP,sizeof(TYP))

/* Simulation der Turbo-Pascal Fillchar-Prozedur */
#define FILLCHAR(VAR, COUNT, C) memset(((char *) VAR), (C), (COUNT));

/* Initialisieren einer Variablen mit einem beliebigen Wert, wobei
** VAR die Adresse der Variablen beschreiben muß */
#define INITVAR(VAR, C) memset(((char *) VAR), (C), sizeof(*VAR));
/*////////////////////////////////////////////////////////////////////////////
///                          Ende der Definitionen                          ///
////////////////////////////////////////////////////////////////////////////*/
```

Bild 2. Allgemein verwendbare Makro-Definitionen erleichtern das Programmieren

Im Falle eines erneuten Interrupt 9H während der Verarbeitung der im BIOS-Puffer befindlichen Zeichen, wird dieser lediglich an die Original-Routine des BIOS weitergereicht. Dieser Tastendruck geht nicht verloren, sondern führt höchstens zu einem weiteren Eintrag im BIOS-Tastaturpuffer, der anschließend von der noch voll im Gange befindlichen Routine MyIsr9 abgearbeitet wird.

Bei sehr schnellem Schreiben könnte es nun passieren, daß es trotz der Kompaktheit von MyIsr9 und KbdQueHandler() zum „Auflaufen" einiger Tastencodes im BIOS-Puffer kommt. Es ist dann nicht mehr sichergestellt, daß das Shift-Statuswort zum Zeitpunkt der Übergabe an KbdQueHandler() den Zustand zum Zeitpunkt des Tastendruckes widerspiegelt. Ich möchte allerdings zu bedenken geben, daß besondere Tastenkombinationen wie z. B. „Shift-Ctrl-Pfeil rechts" in der Regel nicht sehr schnell hintereinander eingegeben werden können und während der Entgegennahme von Fließtext (z. B. in einem Editor) der Shift-Status völlig unerheblich ist. Was die Details der Realisierung anbelangt, verweise ich auf den Quellcode in *Bild 2*, der ausführlich kommentiert und sehr einfach gehalten ist. Aus Gründen der besseren Lesbarkeit des Quelltextes verwende ich auch in C-Programmen sowohl Groß- als auch Kleinbuchstaben. Um den Assembler dazu zu bewegen, sich dieser Angewohnheit zu beugen und seinerseits Symbole abhängig von Groß- und Kleinschreibung zu behandeln (case sensitiv), ist das Modul KBISR mit der Assembleroption /MX zu übersetzen. Der Befehl lautet:

```
MASM /MX kbisr;
```

Außer der Interrupt-Service-Routine MyIsr9 enthält das Modul KBISR.ASM die Funktionen KbdQueHandler(), InitKbdSys() und EndKbdSys(). Die Funktion InitKbdSys() hängt die neue Interrupt-Service-Routine in den Interrupt 9H ein und stellt die Adresse eines privaten Stack für den Queue-Handler bereit. Die Adresse der Stack-Obergrenze wird als Zeiger (Far-Character-Pointer) auf dem Stack erwartet. Das Gegenstück zu InitKbdSys() heißt EndKbdSys(). Diese Funktion stellt den ursprünglichen Zustand wieder her und deaktiviert die eigene Interrupt-Service-Routine. EndKbdSys() muß vor dem Verlassen des Anwendungsprogrammes unbedingt aufgerufen werden, sonst ist ein Systemabsturz unvermeidlich.

Ohne C geht nichts

Außer Assembler-Routinen besteht das Tastatur-Subsystem noch aus dem C-Modul LLKBIO.C (*Bild 3*), der Header-Datei LLKBIO.H (*Bild 4*) und einer Datei mit Makro-Definitionen (sysmac.h). Alle Dateien sind übersichtlich gegliedert und reichhaltig dokumentiert. Zugunsten der leichteren Verständlichkeit ist der C-Code mehr auf Übersichtlichkeit als auf Kompaktheit hin ausgerichtet.

```
/*////////////////////////////////////////////////////////////////////////
///                 Low-Level Keyboard Unterstützung V1.00               ///
///                 Typ- und Datendefinitionen, Macros                   ///
///       Modul..:  LLKBIO.H                                             ///
///       Autor..:  Christian Philipps                                   ///
///       System.:  MS-DOS 3.2 / Microsoft Quick-C/Optimizing C V5.0     ///
///       Stand..:  4/88                                                 ///
////////////////////////////////////////////////////////////////////////*/

/* Defines */
#define BIOS_KEYCODE     unsigned long          /* aus Dokumentationsgründen */
#define NEXTSLOT(P)      (((P)+1)%KbdQueueSize) /* Head-/Tailpointer berechnen */
#define KBD_STACKSIZE    500                 /* 0,5 KB Stack für KbdQueHandler */
#define WAIT             0                    /* warte auf Tastendruck */
#define NOWAIT           1                    /* warte nicht ...  */

/* Hier nun die Defines zum ANDen des Shift-Status (Status-Byte 1) */
#define INS_STATE        0x0080
#define CAPS_STATE       0x0040
#define NUM_STATE        0x0020
#define SCROLL_STATE     0x0010
#define ALT_SHIFT        0x0008
#define CTRL_SHIFT       0x0004
#define LEFT_SHIFT       0x0002
#define RIGHT_SHIFT      0x0001
/* Status-Byte 2 */
#define INS_SHIFT        0x8000
#define CAPS_SHIFT       0x4000
#define NUM_SHIFT        0x2000
#define SCROLL_SHIFT     0x1000
#define LEFT_CTRL_DOWN   0x0100
#define LEFT_ALT_DOWN    0x0200
#define RIGHT_CTRL_DOWN  0x0400                 /* OS/2 */
#define HOLD_STATE       0x0800
#define RIGHT_ALT_DOWN   0x0800                 /* OS/2 */

/* Macros zum Abfragen einzelner Shift-Statusbits */
#define IsIns            (*BIOSShiftStatus & INS_STATE)
#define IsCaps           (*BIOSShiftStatus & CAPS_STATE)
#define IsNum            (*BIOSShiftStatus & NUM_STATE)
#define IsScroll         (*BIOSShiftStatus & SCROLL_STATE)
#define IsAlt            (*BIOSShiftStatus & ALT_SHIFT)
#define IsCtrl           (*BIOSShiftStatus & CTRL_SHIFT)
#define IsLeftShift      (*BIOSShiftStatus & LEFT_SHIFT)
#define IsRightShift     (*BIOSShiftStatus & RIGHT_SHIFT)
#define IsInsShift       (*BIOSShiftStatus & INS_SHIFT)
#define IsCapsShift      (*BIOSShiftStatus & CAPS_SHIFT)
#define IsNumShift       (*BIOSShiftStatus & NUM_SHIFT)
#define IsScrollShift    (*BIOSShiftStatus & SCROLL_SHIFT)
#define IsLeftCtrlDown   (*BIOSShiftStatus & LEFT_CTRL_DOWN)
#define IsRightCtrlDown  (*BIOSShiftStatus & RIGHT_CTRL_DOWN)
#define IsLeftAltDown    (*BIOSShiftStatus & LEFT_ALT_DOWN)
#define IsRightAltDown   (*BIOSShiftStatus & RIGHT_ALT_DOWN)
#define IsHoldState      (*BIOSShiftStatus & HOLD_STATE)

/* Macros zum Setzen einzelner Shift-Statusbits */
#define SetIns           (*BIOSShiftStatus |= INS_STATE)
#define SetCaps          (*BIOSShiftStatus |= CAPS_STATE)
#define SetNum           (*BIOSShiftStatus |= NUM_STATE)
#define SetScroll        (*BIOSShiftStatus |= SCROLL_STATE)
#define SetAlt           (*BIOSShiftStatus |= ALT_SHIFT)
#define SetCtrl          (*BIOSShiftStatus |= CTRL_SHIFT)
#define SetLeftShift     (*BIOSShiftStatus |= LEFT_SHIFT)
#define SetRightShift    (*BIOSShiftStatus |= RIGHT_SHIFT)
#define SetHoldState     (*BIOSShiftStatus |= HOLD_STATE)

/* Macros zum Zuruecksetzen einzelner Shift-Statusbits */
#define ResetIns         (*BIOSShiftStatus &= INS_STATE)
#define ResetCaps        (*BIOSShiftStatus &= CAPS_STATE)
#define ResetNum         (*BIOSShiftStatus &= NUM_STATE)
#define ResetScroll      (*BIOSShiftStatus &= SCROLL_STATE)
#define ResetAlt         (*BIOSShiftStatus &= ALT_SHIFT)
#define ResetCtrl        (*BIOSShiftStatus &= CTRL_SHIFT)
#define ResetLeftShift   (*BIOSShiftStatus &= LEFT_SHIFT)
#define ResetRightShift  (*BIOSShiftStatus &= RIGHT_SHIFT)
#define ResetHoldState   (*BIOSShiftStatus &= HOLD_STATE)

/* Typdefinition für einen BIOS-Tastendruck */
#pragma pack(1)

struct KeyData {                                /* Subset der OS/2-Struktur */
    unsigned char  char_code;
    unsigned char  scan_code;
    unsigned short shift_status;
};

typedef union bios_keytype {
  struct KeyData BIOSKey;
  BIOS_KEYCODE BIOSLongKey;
} BIOS_KEYTYPE;
#pragma pack(2)

/*////////////////////////////////////////////////////////////////////////
///                    E N D   O F   M O D U L E                         ///
////////////////////////////////////////////////////////////////////////*/
```

Bild 3. Header-Datei zur Tastaturabfrage

In der Header-Datei steht die Definition des Typs BIOS_KEYTYPE, die dafür sorgt, daß man das je Tastendruck gewonnene Doppelwort sowohl als BIOS_KEYCODE (ein Synonym für „unsigned long") als auch als Struktur ansprechen kann. Die Struktur KeyData ist eine Untermenge der gleichnamigen Struktur von OS/2, die sich in der OS/2-Header-Datei subcalls.h befindet. Über sie kann man Scan-Code, Zeichen und Shift-Status einzeln ansprechen.

Weiterhin befindet sich in der Datei LLKBIO.H eine Reihe von Konstanten-Definitionen und Makros, mit deren Hilfe man direkt auf das BIOS-Shift-Statuswort zugreifen kann. Die Definitionen sind selbsterklärend und ermöglichen es, beliebige Statusbits abzufragen und, falls es sinnvoll ist, auch zu setzen bzw. zurückzusetzen.
Die Funktion KbdQueHandler(), die in der Datei LLKBIO.C steht, wird von der Interrupt-Service-Routine bei jedem auswertbaren Tastendruck aufgerufen. Sie füllt das Element des Tastatur-Puffers, auf das der Head-Pointer gerade zeigt, mit dem von der Interrupt-Service-Routine auf den Stack gelegten Doppelwort. Anschließend überprüft sie, ob beim Vorwärtsbewegen des Head-Pointers der Puffer überläuft, d. h. Head- und Tail-Pointer auf das gleiche Pufferelement zeigen. Ist das der Fall, wird das Überlaufkennzeichen gesetzt und der Head-Pointer nicht verändert. Anderenfalls wird lediglich der Head-Pointer auf das nächste Pufferelement positioniert und anschließend die Kontrolle an die Interrupt-Service-Routine zurückgegeben. Beachten Sie bitte, daß das Compiler-Pragma check–stack() verwendet wurde, um für den Queue-Handler die Stack-Überlauf-Prüfung abzuschalten. Wird dies unterlassen, kommt es beim ersten Tastendruck nach der Aktivierung des Subsystems zu einem Programmabbruch mit nachfolgendem Systemabsturz. Selbstverständlich kann das Pragma entfallen, wenn man mit dem Compiler-Schalter -Gs den Stack-Check abschaltet.

Ein neues Tastatur-Subsystem

Die Aktivierung des Tastatur-Subsystems wird von der Funktion InstallKbdSubSys() übernommen, die auch auf dem Heap einen privaten Stack für den Queue-Handler anlegt.
Zum Deaktivieren der Interrupt-Routine und Freigeben des vom Stack belegten Speicherplatzes dient die Funktion ShutdownKbdSubSys(). Sie verwendet die Funktion EndKbdSys() und muß aus den bereits genannten Gründen vor dem Verlassen des Programmes aufgerufen werden.

```
/*////////////////////////////////////////////////////////////////////////////
///                Low-Level Keyboard Unterstützung V1.00                   ///
///      Modul..:  LLKBIO.C                                                 ///
///      Autor..:  Christian Philipps     Stand..:   4/88                   ///
///      System.:  MS-DOS 3.2 / Microsoft Quick-C/Optimizing C V5.0         ///
////////////////////////////////////////////////////////////////////////////*/

#include <stdio.h>
#include <stdlib.h>
#include <malloc.h>
#include <sysmac.h>
#include "llkbio.h"

/* Deklarationen der in kbisr.asm definierten Daten und Funktionen.
** KBISR.ASM dient zum Abfangen jeglicher Tastendrücke und zum
** Aufruf der C-Funktion KbdQueHandler für jede über den INT 16H ausles-
** bare Taste. Neben dem Keycode den der INT 16H liefert, werden
** jedoch auch noch der Shift-Status und der erweiterte Shift-Status an
** den Handler übergeben.
** Weiterhin ist im Assemblermodul die Low-Level-Queue definiert. */
extern void InitKbdSys(char *);                 /* Initialisiere Subsystem */
extern void EndKbdSys(void);                    /* Setze INT 9 Vektor zurück */
extern unsigned short KbdQueueSize;             /* Anzahl Queue-Elemente */
extern unsigned short KbdQueueHead;             /* Head-Pointer */
extern unsigned short KbdQueueTail;             /* Tail-Pointer */
extern BIOS_KEYTYPE   KbdQueue;                 /* Erstes Element der Queue */

/* globale Daten */
BIOS_KEYTYPE       *KbdQueuePtr = &KbdQueue;    /* Zeiger zum Queue-Zugriff */
BOOLEAN            KbdOverflow  = FALSE;        /* Pufferüberlauf-Flag */
unsigned short far *BIOSShiftStatus = (unsigned short far *)0x00400017;
char               *KbdSysStack;                /* Zeiger auf den INT 9-Stack */
/*--------------------------------------------------------------------------*/
#pragma check_stack(off)
void KbdQueHandler(Key)
BIOS_KEYCODE Key;
/* Diese Funktion wird nach Aktivierung des Keyboard-Subsystems für
** jedes über die Tastatur eingegebene Zeichen aufgerufen.
** Der Parameter Shift-Status enthält beide BIOS-Statusbytes; im Low-Byte
** das auch über den INT 16H zugängliche Statusbyte und im High-Byte das-
** jenige Statusbyte, das darüber Aufschluß gibt, ob die Toggle-Keys gerade
** heruntergedrückt sind oder nicht und ob gerade die Bildschirmausgabe
** über Ctrl-NumLock abgeschaltet wurde.
** Der Keypress-Parameter enthält im Low-Byte einen ASCII-Code und
** im High-Byte den Scan-Code der Taste. Ist der ASCII-Code 0, so enthält
** das High-Byte den erweiterten Tastencode.
** Bei sehr schnellem Schreiben kann es sein, daß der Shift-Status nicht
** dem Zustand zum Zeitpunkt des Tastendruckes entspricht, da u. U. der
** INT 9H-Handler schneller hintereinander aufgerufen wird, als die Tasten-
** codes über den INT 16H abgeholt und verarbeitet werden können.
** In der Regel dürfte dies kein Hindernis sein, da explizit ausgelöste
** Sondertasten meist nicht sehr schnell hintereinander betätigt werden.
** Der übergebene Tastendruck wird in den Slot gesteckt, der durch
** den HeadPointer adressiert wird. Nachfolgend wird der HeadPointer
** um eine Position weiterbewegt falls dadurch nicht ein Pufferüberlauf
** erzeugt wird.
** Tritt ein Pufferüberlauf auf, so wird das Flag KbdOverflow gesetzt.*/
{
  unsigned short z;

  KbdQueuePtr[KbdQueueHead].BIOSLongKey=Key;
  if((z=NEXTSLOT(KbdQueueHead))==KbdQueueTail)  /* Überlauf */
    KbdOverflow = TRUE;
  else
    KbdQueueHead = z;
}
#pragma check_stack(on)
/*--------------------------------------------------------------------------*/
```

Bild 4. Die C-Funktionen

```
BOOLEAN BIOSKeyAvail()
/* Prüfe, ob in der KbdQueue Tastenanschläge vorliegen und liefere
** TRUE, falls dies der Fall ist. */
{
  return(KbdQueueHead==KbdQueueTail ? FALSE : TRUE);
}
/*--------------------------------------------------------------------------*/
BIOS_KEYCODE BIOSGetKey()
/* Holen eines Tastendruckes aus dem Ringpuffer.
** Es wird vorausgesetzt, daß ein Tastencode vorliegt!!! */
{
  BIOS_KEYTYPE k;
  k.BIOSLongKey=KbdQueuePtr[KbdQueueTail].BIOSLongKey;
  KbdQueueTail=NEXTSLOT(KbdQueueTail);
  return(k.BIOSLongKey);
}
/*--------------------------------------------------------------------------*/
void BIOSPurgeQueue()
/* Löschen des Ringpuffers und Zurücksetzen des Overflow-Merkers */
{
  KbdQueueTail=KbdQueueHead;
  KbdOverflow=FALSE;
}
/*--------------------------------------------------------------------------*/
BOOLEAN AltOnly()
/* Liefert TRUE, wenn nur die ALT-Taste gedrückt und wieder losgelassen
** wurde. Liegen im Tastaturpuffer bereits Tastenanschläge vor, oder
** es wird eine ALT-? Kombination eingegeben, so wird die Kontrolle so-
** fort an den Aufrufer zurückgegeben und die Funktion liefert FALSE. */
{
  BOOLEAN BIOSKeyAvail(void);

/* Sollte die ALT-Taste bereits gedrückt sein, so warten wir zunächst
** darauf, daß diese losgelassen wird oder ein Tastendruck im Tastatur-
** puffer landet. */
  while(IsAlt && !BIOSKeyAvail())               /* wieder warten */
  ;
/* Liegt nun bereits etwas im Tastaturpuffer herum, so können wir den Traum
** von der alleine gedrückten ALT-Taste getrost vergessen */
  if(BIOSKeyAvail())
    return(FALSE);

/* Warten auf Tastendruck oder auf ALT-Taste */
  while(!(IsAlt || BIOSKeyAvail()))
  ;
/* Liegt nun noch immer keine Taste im Puffer, so wird solange gewartet,
** bis ein Tastendruck anliegt oder die ALT-Taste losgelassen wurde. */
  if(BIOSKeyAvail())                            /* Taste da, also zurück */
    return(FALSE);
  while(IsAlt && !BIOSKeyAvail())               /* wieder warten */
  ;
/* Der Funktionswert entspricht nun TRUE, falls noch immer kein Tasten-
** anschlag vorliegt und FALSE, wenn jetzt der Tastaturpuffer nicht mehr
** leer ist. */
  return(!BIOSKeyAvail());
}

/*--------------------------------------------------------------------------*/
void KbdCharIn(KeyBuf, WaitFlag, KbdHandle)
struct KeyData *KeyBuf;
int WaitFlag, KbdHandle;
/* Versuch der Imitation der gleichnamigen OS/2-Funktion. Der Parameter
** KbdHandle dient als Platzhalter für einen weiteren Parameter, der unter
** OS/2 gefordert ist; hier hat er keine Funktion.
** Aufgabe: Abholen eines Zeichens von der Tastatur sofern eines vorhanden
** ist. Ist WaitFlag WAIT, so wird auf einen Tastendruck gewartet wenn
** gerade keiner ansteht. Anderenfalls wird als Zeichen und Scan-Code
** jeweils 0 geliefert, jedoch zusätzlich der aktuelle Shift-Status er-
** mittelt. */
{
```

```
  BOOLEAN BIOSKeyAvail(void);
  BIOS_KEYCODE BIOSGetKey(void);

/* Falls Warten angefordert ist, dann warte auf Tastendruck */
  while(WaitFlag==WAIT && !BIOSKeyAvail())
  ;
/* Liegt ein Tastendruck an, so wird dieser in den Puffer dessen Adresse
** als Parameter übergeben wurde übertragen. Anderenfalls werden Scan-
** Code und Character auf 0 gesetzt und der Shift-Status ermittelt. */
  if(BIOSKeyAvail()) {
    *((BIOS_KEYCODE *)KeyBuf) = BIOSGetKey();
  }
  else {
    KeyBuf->scan_code    = 0;
    KeyBuf->char_code    = 0;
    KeyBuf->shift_status = *BIOSShiftStatus;
  }
}
/*---------------------------------------------------------------------------*/
void KbdPeek(KeyBuf, KbdHandle)
struct KeyData *KeyBuf;
int KbdHandle;
/* Versuch der Imitation der gleichnamigen OS/2-Funktion. Der Parameter
** KbdHandle dient als Platzhalter für einen weiteren Parameter, der unter
** OS/2 gefordert ist; hier hat er keine Funktion.
** Aufgabe: Liefert das erste Element der Keyboard-Queue, so eines
** vorhanden ist, entfernt dieses jedoch nicht aus der Warteschlange */
{
  BOOLEAN BIOSKeyAvail(void);

/* Liegt ein Tastendruck an, so wird dieser in den Puffer dessen Adresse
** als Parameter übergeben wurde übertragen. Anderenfalls werden Scan-
** Code und Character auf 0 gesetzt und der Shift-Status ermittelt. */
  if(BIOSKeyAvail()) {
    *((BIOS_KEYCODE *)KeyBuf) = KbdQueuePtr[KbdQueueTail].BIOSLongKey;
  }
  else {
    KeyBuf->scan_code    = 0;
    KeyBuf->char_code    = 0;
    KeyBuf->shift_status = *BIOSShiftStatus;
  }
}
/*---------------------------------------------------------------------------*/
BOOLEAN InstallKbdSubSys()
/* Anlegen eines Stacks für das Keyboard-Subsystem und installieren einer
** eignenen Interruptroutine für den Interrupt 9.
** Die Funktion liefert den Wert FALSE, sollte das Anlegen des Stacks auf
** dem Heap nicht erfolgreich verlaufen. * /
{
  extern void InitKbdSys(char *);

  if((KbdSysStack=CALLOC(char, KBD_STACKSIZE))==NULL_CP)
    return(FALSE);

  InitKbdSys(KbdSysStack+KBD_STACKSIZE);        /* ISR installieren */
  return(TRUE);
}
/*---------------------------------------------------------------------------*/
void ShutdownKbdSubSys()
/* Deinstallieren der INT 9 Service-Routine und freigeben des Systemstacks */
{
  extern void EndKbdSys();

  EndKbdSys();
  free(KbdSysStack);
}
/*////////////////////////////////////////////////////////////////////////////
///                    E N D   O F   M O D U L E                           ///
////////////////////////////////////////////////////////////////////////////*/
```

```
/*////////////////////////////////////////////////////////////////////////////
///                 Low-Level Keyboard Unterstützung V1.00                 ///
///                       D e m o p r o g r a m m                          ///
///      Modul..:   KBIOTEST.C    Stand..:   4/88                         ///
///      Autor..:   Christian Philipps                                     ///
///      System.:   MS-DOS 3.2 / Microsoft Quick-C/Optimizing C V5.0       ///
////////////////////////////////////////////////////////////////////////////*/

#include <stdio.h>
#include <stdlib.h>
#include <sysmac.h>
#include "llkbio.h"
#define ESC 0x1B

main()
{
  BOOLEAN InstallKbdSubSys(void);
  void EndKbdSubSys(void);
  void BIOSPurgeQueue(void);
  BOOLEAN BIOSKeyAvail(void);
  BIOS_KEYCODE BIOSGetKey(void);
  void KbdCharIn(struct KeyData *, int, int);
  void KbdPeek(struct KeyData *, int);

  struct KeyData key;
  int n;

/* Keyboard-Subsystem aktivieren */
  if(!InstallKbdSubSys()) {
    puts("Fehler bei der Stackzuordnung!!");
    exit(1);
  }
  puts("O.K. - Ich warte auf Tastendrücke; Ende mit ESC!\n");

/* Verarbeitungsschleife bis ESC gedrückt wurde */
  do {
/* Nur Alt gedrückt */
    if(AltOnly()) {
      puts("ALT-Taste alleine gedrückt und wieder losgelassen!");
    } /*if*/
/* Tastencode im Puffer angekommen */
    else {
      KbdCharIn(&key,NOWAIT,0);
      printf("ASCII-Code: %3d / 0x%2x  Scan-Code: %3d / 0x%2x  Shift-Status:
0x%4x\n",
              key.char_code, key.char_code, key.scan_code, key.scan_code,
              key.shift_status);
    } /*else*/
  } while(key.char_code != ESC);

/* Keyboard-Subsystem deinstallieren */
  ShutdownKbdSubSys();
  puts("Das war's!\n");
  exit(0);
}
```

Bild 5. Ein kleines Demo-Programm

Die Funktion BIOSKeyAvail() stellt fest, ob der Puffer Zeichen enthält. Sie liefert den Wert TRUE, falls mindestens ein Tastendruck ansteht. Stehen Tastendrücke an, und das sei hier vorausgesetzt, können diese als Werte vom Typ ...BIOS_ICEYCODE mittels BIOSGetKey() ausgelesen werden. Alle Einträge im Tastatur-Puffer löscht die Funktion BIOSPurgeQueue().

Wesentlich interessantere Funktionen, die sogar ein Gegenstück unter OS/2 haben, sind KbdCharIn() und KbdPeek(). Letztere Funktion liest das erste Element aus dem Tastatur-Puffer ohne es zu löschen.

Beide Funktionen erwarten als ersten Parameter die Adressse einer Struktur vom Typ KeyData. Unter OS/2 enthält diese allerdings weitaus mehr Elemente als unter DOS. Die Strukturelemente, die von den in LLKBIO.C codierten Funktionen gefüllt werden, tragen die gleichen Namen wie unter OS/2, so daß Programme, die mit diesen Routinen erstellt wurden, im Hinblick auf die Tastaturabfrage auch unter OS/2 lauffähig sein müßten. Der letzte Parameter beider Funktionen ist ein Platzhalter, der unter DOS keine Funktion hat. An seiner Stelle wird unter OS/2 das Handle der angesprochenen logischen Tastatur erwartet. Beim Aufruf der Funktion KbdCharIn() kann man durch den zweiten Parameter angeben, ob sie auf einen Tastendruck warten soll, wenn gerade keine Tastenanschläge zur Bearbeitung anstehen. Zulässige Werte sind die Konstanten WAIT oder NOWAIT. Lautet der zweite Parameter WAIT, wartet die Funktion bei leerem Tastatur-Puffer auf eine Betätigung irgendeiner Taste.

Die Funktion KbdPeek() kehrt grundsätzlich sofort zurück und liefert die gleichen Werte, die auch KbdCharIn() zurückreicht, wenn kein Warten auf eine Taste gefordert wurde. Ist der Tastatur-Puffer zum Zeitpunkt des Aufrufs leer, werden Zeichen und Scan-Code auf 0 gesetzt. Der aktuelle Shift-Status wird jedoch ordnungsgemäß ermittelt. Unter OS/2 läßt sich z. B. mit dieser Funktion feststellen, ob ein Tastendruck zur Bearbeitung anliegt oder nicht.

Eine weitere wichtige Funktion heißt AltOnly(). Sie stellt fest, ob nur die Alt-Taste gedrückt und wieder losgelasssen wurde. Hiermit ist das eingangs beschriebene Verhalten des Quick-C-Editors nachbildbar. AltOnly() kehrt erst dann wieder zurück, wenn entweder ein Tastencode im Tastatur-Puffer vorliegt oder die Alt-Taste gedrückt und wieder losgelassen wurde. Im letzteren Fall ist der Tastatur-Puffer nach der Rückkehr der Funktion nach wie vor leer und AltOnly() liefert den Wert TRUE. Anderenfalls wird als Funktionswert FALSE zurückgeliefert. Dies ist auch bei Tastenkombinationen wie z. B. Alt-F1 der Fall.

Ein kleines Demo-Programm, das sowohl den ASCII-Code einer Taste als auch den Scan-Code und den Status der Shift-Taste ausgibt, ist in *Bild 5* dargestellt.

Kompatibel zu OS/2

Die vorgestellten Funktionen dürften auch für Programmierer interessant sein, die beabsichtigen schon heute ihre Programme soweit wie möglich auf eine spätere Portierung auf OS/2 auszurichten. Durch die OS/2-ähnlichen Aufrufe dürften bei der Verwendung dieser Funktionen nur geringe Anpassungen im Bereich der Tastaturbehandlung erforderlich sein.

Das C-Modul LLKBIO.C wurde mit dem Quick-C-Compiler von Microsoft entwikkelt und muß mit der Option -AL (large model) compiliert werden. Die Wahl des Speichermodells ist durch den Kontext begründet, in dem diese Routinen bei mir im Einsatz sind. Soll ein anderes Speichermodell verwendet werden, sind einige Anpassungen im Assemblerteil vorzunehmen. □

Mehrere DOS-Kommandos auf einen Streich

Unter MS-DOS ist es nicht möglich, mehrere Befehle in einer Kommandozeile einzugeben. Eine Ausnahme bilden Pipe-Befehle. So gibt z. B. der Befehl dir | sort | more ein sortiertes Datei-Verzeichnis seitenweise aus. Mit dem kleinen Hilfsprogramm Batcreate kann man mehrere durch Semikolon oder Doppelkreuz (#) getrennte Befehle unter DOS eingeben. Man kopiert z. B. eine Datei von einem Verzeichnis in ein anderes und löscht die Datei im alten Verzeichnis mit einem einzigen Kommando. So sieht der Befehl aus: x copy \versuch.txt \word\versuch.txt; del versuch.txt Das aufrufende Programm besteht aus der Batch-Datei X.bat, mit den Einträgen Batcreate und Job. X.bat ruft zuerst Batcreate auf. Aus der Kommandozeile von X.bat erzeugt Batcreate die Batch-Datei Job.bat. Danach wird Job.bat gestartet und die eingegebenen Befehle ausgeführt. *Michael Lehning*

```
program Batch_Creater;
const
     Return= ^M^J;              { Carriage-Return + Linefeed }
type
     Tname = string [255];
Const
     First_ofs = $82;           { Alten Command-String lesen, der von X.bat
                                  aufgerufen wurde. }
var Command_Laenge,             { Laenge von Command_Str }
    i                : integer;
    Datei            : text;
    Buchstabe        : char;
    Command_Str      : tname;

Procedure Exchange (var Source:tname;Position,Del_Length:byte; Obj: tname);

Begin                                   { Löscht aus Source den Bereich von }
Delete (Source, Position, Del_Length);{ Position bis Position + Del_Length}
Insert (Return, Source, Position);    { und fügt dann dort Obj ein.        }
End; { of procedure Exchange }

begin
  Command_Laenge := 0; Command_Str :=  '';    { Init der Variablen }
  Repeat
   Command_str := Command_str + chr (mem [Cseg:First_ofs+Command_Laenge]);
   Command_Laenge := Command_Laenge+1;
  until Command_Str [Command_Laenge] in  [^M,^@]; { Return oder chr (0) }

  if Command_Laenge > 0 then
  for i := Command_Laenge downto 1 do
      begin
         if Command_str [i] = '#' then Exchange (Command_str,i,1,Return);
         if Command_Str [i] = ';' then
             if copy (Command_str,i+1,2) <> Return
                then Exchange (Command_str,i,1,Return);
      end;  { of Downto-Loop i }
 Assign (datei,'Job.Bat');
 rewrite (datei);                { Schreibe Kommando-String    }
 writeln (Datei, Command_str);  { in die Datei Job.bat        }
 close (Datei);
end.
```

Ein Turbo-Pascal-Programm erleichtert die Bedienung von MS-DOS-Computern

Bryan Hayes

Aus .C mach .H

Header-Dateien per Programm erzeugen lassen

Bei größeren Programmen nimmt die spätere Pflege meist mehr Zeit in Anspruch als die eigentliche Erstellung. Da macht es sich bezahlt, wenn man mit Software-Werkzeugen Arbeit sparen kann – etwa mit dem hier vorgestellten Programm, das automatisch Header-Dateien generiert.

Welchem C-Programmierer ist die Situation nicht bekannt: Aus einem kleinen, harmlosen Projekt sind auf einmal mehrere tausend Zeilen geworden, von denen ein Großteil der Einfachheit halber – und weil es sich so ergeben hat – per „include"-Anweisung eingebunden werden. Dabei wäre es ja eigentlich günstiger (und professioneller), die Include-Dateien in Objekt-Dateien zu verwandeln, wodurch der Compilier/Link-Vorgang um einiges schneller vonstatten ginge. Allerdings braucht man dann natürlich auch eine entsprechende Header-Datei, damit das Hauptprogramm (oder auch andere Include-Dateien) Zugriff auf die nötigen '#define'-Anweisungen, globale Variablen und Funktions-Prototypen haben. Nun, anhand der Quelldatei eine Objekt-Datei zu erzeugen, ist kein Problem, wohl aber, anhand der C-Quelle eine Header-Datei zu erstellen. Zum einen deswegen, weil die .C-Datei sich noch mindestens fünfzigmal ändern wird und zum anderen, weil man meist keine besondere Lust verspürt, alle nötigen #defines, globale Variablen und Funktionsprototypen irgendwie herauszuschreiben bzw. zu kopieren.

Das Programm

Genau dafür ist das Programm MH (soll heißen: MAKE .H) in *Bild 1* gedacht: Es erstellt anhand einer .C-Datei automatisch eine entsprechende .H-Datei. Allerdings kommt man nicht umhin, anzugeben, was in der .H-Datei erscheinen soll; es sind dazu lediglich ein paar kleine Kommentare nötig, die dann von MH ausgewertet werden.

MH wertet folgende Kommentare aus, die jeweils in einer eigenen Zeile stehen müssen:

1. /**f*/
   Dieser Kommentar steht vor einer Funktion. Anhand der Zeilen
   */**f*/*
   *void *ff(int a, char b)*
   erzeugt MH die Zeile
   *void *ff(int a, char b);*
2. /**e*/
   Dieser Kommentar steht vor einer globalen Variable:
   */**e*/*
   *char *vv[100]= { ...*
   Daraus wird dann
   *extern char *vv[100];*
   Zu beachten ist hierbei, daß nur der Text bis zum ersten Gleichheitszeichen oder Semikolon übertragen wird.
3. /**{*/
   Dieser Kommentar markiert den Beginn eines Blockes von Zeilen, die direkt in

```
 1 #include "stdio.hhh"       /*  printf, ...  */
 2 #include "fcntl.hhh"       /*  O_xxx */
 3 #include "io.hhh"          /*  open, read, ...  */
 4 #include "string.hhh"      /*  strxxx  */
 5 #include "process.hhh"     /*  exit  */
 6
 7 #define BYTE char          /*  8 bit  */
 8 #define UBYTE unsigned char
 9 #define WORD int           /*  16 bit */
10 #define UWORD unsigned int
11 #define LONG long          /*  32 bit  */
12 #define ULONG unsigned long
13
14
15
16 /**{*/
17 #define EOL 10
18
19 #define InBufferSize 0x4000
20 /**}*/
21
22 struct { UBYTE b[2]; } EOLString={ 0xD, 0xA };
23
24 /**e*/
25 UBYTE InBuffer[ InBufferSize ];
26 /**e*/
27 UWORD InBufferCnt=0;
28
29 #define OutBufferSize 0x4000
30 UBYTE OutBuffer[ OutBufferSize ];
31 UWORD OutBufferCnt=0;
32
33 ULONG OutNrBytes=0;
34
35 WORD InFileNr, OutFileNr;
36 UBYTE InEOF=0;
37
38 UBYTE ParN, ParQ;
39
40
41
42 UBYTE RB( void )     /*  read 1 byte  */
43 {
44   register UBYTE b;
45   register UWORD len;
46
47   if( InEOF )
48     return( (UBYTE)0 );
49
50   if( InBufferCnt == 0 )
51   {
52     if( (len=read( InFileNr, InBuffer,
53     (UWORD)InBufferSize )) < InBufferSize )
54       InBuffer[ len ]=0;
55   }
56   InEOF=!(b=InBuffer[ InBufferCnt++ ]);
57   if( InBufferCnt == InBufferSize )
58     InBufferCnt=0;
59   return( b );
60 }
61
62
63
64 void WB( UBYTE b )     /*  write 1 byte  */
```

Bild 1. Das Programm MH.C ist leicht auf verschiedene C-Compiler (und damit verschiedene Computer) zu übertragen

```
{
  register UBYTE *p;
  register UWORD z;

  if( b == 0 )
  {
    p=(UBYTE *)&EOLString;
    z=sizeof( EOLString );
    do
      WB( *p++ );
    while( --z );
  }
  else if( b >= 2 )
    OutBuffer[ OutBufferCnt++ ]=b;
  if( OutBufferCnt == OutBufferSize  ||  b == 1 )
  {
    if( OutBufferCnt != 0 )
    {
      write( OutFileNr, OutBuffer, OutBufferCnt );
      OutNrBytes += OutBufferCnt;
      if( !ParQ )
      {
        p=OutBuffer;
        z=OutBufferCnt;
        do
          putchar( *p++ );
        while( --z );
      }
      OutBufferCnt=0;
    }
  }
}


/**f*/
void WS( UBYTE *s )    /*  write 1 string  */
{
  register UBYTE b;

  while( (b=*s++) != 0 )
    WB( b );
}


UBYTE FindMark( void )    /*  find mark / * * x * /  */
{
  register UBYTE foundmark, b;

  foundmark=0;
  do
  {
    while( RB() != '/'  &&  !InEOF );    /*  find /  */
    if( RB() == '*' )
    if( RB() == '*' )
    if( (b=RB()) != 0 )
    if( RB() == '*' )
    if( RB() == '/' )
    {
      foundmark=1;
      while( RB() != EOL  &&  !InEOF );    /*  find EOL  */
    }
  }
  while( !foundmark  &&  !InEOF );
  return( b );
}


void CopyFunction( void )
{
  register UBYTE NrBrackets, b;

  NrBrackets=0;
  do
  {
    b=RB();
    if( b == '(' )
    {
      NrBrackets++;
      if( ParN )
      {
        WB( b );
        b=')';
      }
    }
    if( b == ')' )
      NrBrackets--;
    if( b >= ' ' )
      WB( b );
    else if( b == EOL )
      WB( (UBYTE)0 );
  }
  while( !InEOF  &&  !(b == ')'  &&  NrBrackets == 0) );
  WB( ';' ); WB( (UBYTE)0 );
}


void CopyExtern( void )
{
  register UBYTE b;

  WS( "extern " );
  do
  {
    b=RB();
    if( b == EOL )
      WB( (UBYTE)0 );
    else if( b >= ' '  &&  b != '='  &&  b != ';' )
      WB( b );
  }
  while( !InEOF  &&  b != '='  &&  b != ';' );
  WB( ';' ); WB( (UBYTE)0 );
}

```

```
182
183
184 void CopyBlock( void )
185 {
186   register UBYTE b;
187
188   do
189   {
190     b=RB();
191     if( b == '/' )
192     {
193       if( !InEOF )
194       {
195         if( (b=RB()) == '*' )
196         {
197           if( !InEOF )
198           {
199             if( (b=RB()) == '*' )
200             {
201               if( !InEOF )
202               {
203                 if( (b=RB()) == '}' )
204                 {
205                   RB(); RB();    /*  read *, /  */
206                   b=0;
207                 }
208                 else
209                   WS( "/**" );
210               }
211             }
212             else
213               WS( "/*" );
214           }
215         }
216         else
217           WB( '/' );
218       }
219     }
220     if( b == EOL )
221       WB( (UBYTE)0 );
222     else if( b >= ' ' )
223       WB( b );
224   }
225   while( !InEOF  &&  b != 0 );
226 }
227
228
229
230 /**f*/
231 void main(
232   int argc,
233   UBYTE *argv[]
234   )
235 {
236   UBYTE InFileName[ 256 ], OutFileName[ 256 ];
237   UBYTE b;
238
239   printf( "MH (C) 8.2.1988 by Bryan Hayes\n" );
240
241   if( argc == 1 )    /*  if no parameters given  */
242   {
243     printf( "Parameters: filename </n> </q>\n" );
244     printf( "\n" );
245     printf( "If there is no filename extension, .c will be added.\n" );
246     printf( "The option /n suppresses the output of parameters belonging\n"
);
247     printf( "to a function prototype:\n" );
248     printf( "  void *ff( int a, char b ) will then be put out as\n" );
249     printf( "  void *ff();\n" );
250     printf( "The option /q suppresses the echo on the screen.\n" );
251     exit( 1000 );
252   }
253
254   strcpy( InFileName, argv[ 1 ] );
255   if( strchr( InFileName, '.' ) == 0 )
256     strcat( InFileName, ".c" );    /*  standard extension .c  */
257   strcpy( OutFileName, InFileName );
258   strcpy( strchr( OutFileName, '.' ), ".h" );    /*  output extension .h  *
/
259
260   ParN=(*(argv[2]+1) & 0xDF) == 'N'  ||  (*(argv[3]+1) & 0xDF) == 'N';
261   ParQ=(*(argv[2]+1) & 0xDF) == 'Q'  ||  (*(argv[3]+1) & 0xDF) == 'Q';
262
263   if( (InFileNr=open( InFileName, O_BINARY | O_RDONLY )) <= 0 )
264   {
265     printf( "File %s: open error\n", InFileName );
266     exit( 1000 );
267   }
268   if( (OutFileNr=open( OutFileName, O_BINARY | O_RDWR | O_CREAT | O_TRUNC,
269   0xFFFF )) <= 0 )
270   {
271     printf( "File %s: open error\n", OutFileName );
272     close( InFileNr );
273     exit( 1000 );
274   }
275
276   do    /*  main loop  */
277   {
278     b=FindMark();
279     if( !InEOF )
280     {
281       if( b == 'f' )
282         CopyFunction();
283       else if( b == 'e' )
284         CopyExtern();
285       else if( b == '{' )
286         CopyBlock();
287     }
288   }
289   while( !InEOF );
290
291   close( InFileNr );
292   WB( (UBYTE)1 );    /*  write data in buffer  */
293   close( OutFileNr );
294
295   printf( "Made file %s, %ld bytes\n", OutFileName, OutNrBytes );
296 }
297
```

```
 1 #define EOL 10
 2
 3 #define InBufferSize 0x4000
 4 extern UBYTE InBuffer[ InBufferSize ];
 5 extern UWORD InBufferCnt;
 6 void WS( UBYTE *s );
 7 void main(
 8   int argc,
 9   UBYTE *argv[]
10   );
```

Bild 2. Diese Header-Datei MH.H wird erzeugt, wenn man MH auf seinen eigenen Quelltext anwendet

die .H-Datei übertragen werden. Das Blockende muß mit /**}*/ markiert sein:

*/**{*/*
#define CharPerX 80
#define CharPerY 25
*/**}*/*

Die beiden #define-Zeilen werden wie erwähnt direkt übertragen.

In *Bild 2* ist MH.H zu sehen, das sich ergibt, wenn man MH auf MH.C anwendet.

Aufruf

Wird MH ohne Parameter aufgerufen, werden die nötigen Parameter aufgelistet. In jedem Fall muß der Name der .C-Datei angegeben werden, wobei automatisch ein '.C' angehängt wird, falls keine Extension angegeben ist. Die Ausgabedatei (die übrigens immer erzeugt wird, auch wenn sie keinen Inhalt hat) bekommt automatisch die Extension .H. Normalerweise wird der Inhalt der .H-Datei auch auf dem Bildschirm ausgegeben; dies läßt sich aber durch die Option /q unterdrücken. Durch die andere Option /n kann man die Angabe der Parameter bei Funktionsprototypen verhindern, d. h., es wird immer ...(); ausgegeben. Dies ist bei älteren Compilern, die die Datentypen sowieso nicht kontrollieren, sinnvoll, insbesondere, wenn sich der Compiler über die Parameter beschwert. MH sollte eigentlich mit allen C-Compilern übersetzt werden können, zu beachten sind lediglich ein paar Kleinigkeiten:

- UBYTE, WORD usw. müssen richtig definiert werden.
- EOLString enthält die nötige CR-Sequenz, unter MS-DOS also 0xD, 0xA.
- EOL ist der Code, der am Ende der Zeile auftaucht, dies dürfte bei allen üblichen Systemen 0xA sein (auch bei MS-DOS).
- Außerdem muß noch open beachtet werden: Bei Turbo-C muß hier unbedingt O_BINARY angegeben werden, da ansonsten die Sequenz 0xD, 0xA in 0xA umgewandelt wird (die verschwundenen 0xDs verschwinden dabei leider auch bei read: wenn man 4096 Bytes anfordert, werden lediglich etwa 3800 Bytes als gelesen gemeldet, was normalerweise ein Zeichen für das Ende der Datei ist), bei anderen Systemen kann eventuell darauf verzichtet werden.

Der dritte Parameter bei open, mit dem man irgendwelche Attribute der Datei definieren kann, kann auch weggelassen werden. Ich habe ihn unter Turbo-C 1.0 auf 0xFFFF gesetzt, da die laut Handbuch vorhandenen Konstanten (S_IREAD, S_IWRITE: Lesen / Schreiben erlaubt) nicht in fcntl.h definiert sind, obwohl dies im Handbuch so angegeben ist.

Geschwindigkeit

Zur Geschwindigkeit ist zu sagen, daß sie recht hoch ist (das liegt sicher auch an der etwas assemblernahen Programmierweise): Auf einem AT mit Festplatte wurden nur 1,3 Sekunden (mit Option /q) benötigt, um anhand eines 23 KByte langen Quelltextes eine 2,4 KByte lange Header-Datei zu erstellen (die Zeit für das Laden von MH ist hierbei eingeschlossen). Für die entsprechende Operation auf Diskette wurden 3,9 Sekunden benötigt. MH wurde übrigens ursprünglich auf einem Amiga entwickelt (O_BINARY fällt hierbei weg und EOLString wird auf 0xA gesetzt) und hat sich auch bei mittelgroßen Projekten (4000...6000 Zeilen) unter MS-DOS (Turbo-C) bewährt. □

Extrem schnelle lineare Semicustom-Arrays

Für Entwickler, die ultraschnelle Linear-Arrays mit extremen Leistungsmerkmalen benötigen, ergänzte AT&T ihre Semicustom-Familie um die Typen ALA201 und ALA202. Sie werden in bipolarer Komplementärtechnik mit zwei Metallisierungsebenen gefertigt. Das ist die anspruchvollste Technik, die derzeit auf dem Markt verfügbar ist.

Die Linear-Arrays bieten den Vorteil gleicher Charakteristiken der npn- und pnp-Transistoren bei sehr hohen Frequenzen. Die Grenzfrequenz $f_{(T)}$ für den npn-Transistor beträgt 4,5 GHz, für den pnp-Transistor 3,75 GHz.

Unter einem linearen integrierten Semicustom-Schaltkreis versteht man ein vordefiniertes Array von Widerständen, Kondensatoren, Transistoren und integrierten Dioden.

Mit CAD-Programmen, die auch PC-gestützt sein können, lassen sich diese Arrays vom Anwender schnell zu einer Vielzahl analoger Funktionen konfigurieren, wie sie etwa in der Datenerfassung, für den Schaltbetrieb, die Signalaufbereitung, die Video-Signalverarbeitung oder lineare Regelungen benötigt werden.

Die Anwender erstellen mit CAD-Programmen die benötigten Verbindungen und simulieren die Schaltung. AT&T prüft den Entwurf eingehend und entwickelt die Prüftechnik für das Bauelement. In den meisten Fällen kann AT&T dem Kunden innerhalb von acht bis zehn Wochen nach Beendigung des Entwurfs ein geprüftes Chip aushändigen.

Der Typ ALA201 besteht aus fünf Standardmodulen und einem Leistungsmodul; der Typ ALA202 aus neun Standard- und zwei Leistungsmodulen sowie einem Eingangsmodul. Beide sind in ein regelmäßiges Gittersystem eingefügt, das den Verbindungsaufbau erleichtert. Mehrfache Kollektorkontakte optimieren die Ausgangs-Treiberfähigkeit.

Typische Anwendungen sind automatische Testsysteme, Meßgeräte, Grafik-Systeme und System der Nachrichtentechnik.

Nach Unterlagen von AT&T

Marek S. Paszcza

Speicherverwaltungs-techniken

Die Entwicklung der Techniken zur Verwaltung des Hauptspeichers

Mit der Entwicklung immer leistungsfähigerer Computer ging die Entwicklung der Speicherverwaltungstechniken Hand in Hand. Waren es anfangs nur 64 KByte, die für Programme und Daten reichen mußten, so sind es heute mehrere MByte. Dieser Beitrag schildert die Verfahren.

Bei der Beschreibung der heutigen Personal-Computer werden am häufigsten folgende Parameter angegeben: Verarbeitungsbreite, Taktfrequenz, Speichergröße, anschließbarer Massenspeicher usw. Selten wird dabei auf die im System angewendete Technik der Speicherverwaltung hingewiesen. Allerdings gehört dieses Merkmal zu den entscheidenden Leistungsmerkmalen von Computern. Deren Leistung steigt in den Bereich von Millionen Byte von RAM und mehreren MIPS mit 32-Bit-Daten. In diesem Artikel wird die geschichtliche Entwicklung der in den Personal-Computern angewendeten Speicherverwaltungsstrategien beschrieben und analysiert. Zum Schluß werden einige Richtungen der künftigen Entwicklung von Speichersystemen gezeigt.

Über die Wurzeln

In den Anfängen der Personal-Computer-Ära gab es wenige Alternativen für die Lokalisierung der Programme im Hauptspeicher, weil die Kapazität der Hauptspeicher sehr gering war. Im Unterschied zu den damaligen Nutzern der großen Computerzentren machten sich die ersten Computerfreaks wenig Gedanken über die Zeitverluste und die Zwangspausen in Erwartung von irgendeinem Lebenszeichen, die durch die „Verarbeitung" (meistens im Interpretermodus) von zu großen Datenpaketen verursacht wurden. Auch die Leistungsverluste des Computers, wenn nach der glücklichen Befehlsausführung auf den nächsten Wunsch des Anwenders gewartet werden mußte, störte die ersten Freaks wenig. Mit ein paar von damals kostbaren KByte von RAM gab es für Amateure keine Möglichkeit, in der Zwischenzeit mehrere Programme im Speicher zu laden und abzuarbeiten. Die Fortschritte in der Technologie erlaubten schließlich den Durchbruch auf diesem Gebiet, aber die Vorstellungen und Anforderungen der PC-Anwender haben sich auch geändert. Der größere Speicher (letztlich auch mehr als 1 MByte) und 16/32-Bit-Prozessoren steigern wesentlich die Leistungsfähigkeit der Personal-Computer. Zudem werden völlig neue Einsatzgebiete durch schnelle Winchester-Laufwerke mit mehreren MByte erschlossen. Die Merkmale der Festplatten-Laufwerke ermöglichen die Implementation von virtuellen Maschinen mit der entsprechend effektiven Speicherverwaltung in einer Multitasking-Umgebung.

Solche Mechanismen sind bereits seit Jahren bekannt und in größeren Anlagen im Einsatz. Es verging einige Zeit, bevor sie sich auch im PC-Bereich sinnvoll einsetzen ließen. Die ersten Mikroprozessoren mit virtuellen Eigenschaften (Motorola 68010, Zilog Z8003/4) erschienen zwar vor einiger Zeit auf dem Markt, die Entwicklung der Betriebssystemsoftware ging aber nicht im gleichen Rhythmus voran. Die ersten Personal-Computer mit virtueller Speicherverwaltung zu vernünftig angesetzten Preisen basieren auf Motorola-68010/20- und Intel-80386-Prozessoren und erreichen gerade die Marktreife. Es bleibt abzuwarten, welches Betriebssystemkonzept sich als Standard durchsetzen wird.

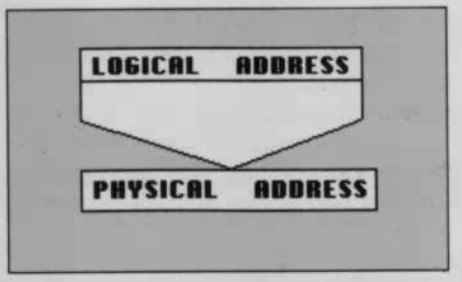


Bild 1. Bei 8-Bit-Systemen stimmen die physischen Speicheradressen mit den logischen Adressen überein

Die 8-Bit-Systeme – CP/M

Die verbreitetsten Betriebssysteme für die 8-Bit-Mikroprozessoren, wie CP/M für die Intel-8080- und Zilog-Z80-Prozessoren, unterstützen einen einfachen aber effektiven Mechanismus zur Datenverarbeitung mit Diskettenlaufwerken als Hintergrundspeicher. Da die 8-Bit-Mikroprozessoren direkt nur einen Speicherbereich von 64 KByte adressieren können, bleiben in typischen Systemen ungefähr 40 KByte für Anwendungsprogramme frei. Der Rest wird vom Betriebssystem selbst, ggf. Video-RAM, Peripherietreiber und -puffer sowie Systemstack verwendet. Die Kommunikation zwischen dem System und den Diskettenlaufwerken wird durch die Teile des Betriebssystems gesichert, die BIOS und BDOS genannt werden. Diese Kommunikation belegt zusätzlich Teile des Hauptspeichers für die Pufferbereiche sowie Floppydeskriptoren.

Die Anwenderprogramme werden in diesen Systemen an festen physischen Adressen gebunden, bei CP/M 0100 hexadezimal. Alle Sprung-, Unterprogrammaufruf- und Datenadressen werden beim Linken fest vereinbart. Die Zusammenarbeit mit den Diskettenlaufwerken läuft nach dem Pufferverfahren ab. Ein Puffer ist ein fixierter Speicherbereich mit einem entsprechenden Deskriptor, in dem die Daten, statt gleich auf die Diskette geschrieben bzw. von der Floppy gelesen zu werden, vorläufig gespeichert werden, um unötiges Hin- und Herladen zu sparen (*Bild 1*).

Wenn aber der von der Datei benötigte Speicherbereich größer ist als der des Puffers, was häufig vorkommt, hat der Programmierer eine Menge zu tun, um die Managementprobleme bei der Aktualisierung der Datei (mit Hilfe von temporären Dateien bzw. durch das ständige Ausladen der Pufferdaten auf die Diskette) zu lösen.

Um das Problem zu verdeutlichen, schauen wir uns das Beispiel eines Textprozessors an. Die typische Sektorgröße einer Diskette beträgt bei CP/M 512 Byte. Die Puffergröße beträgt meistens ein Vielfaches davon, kann aber die angegebene Größe des adressierbaren Speicherbereiches nicht überschreiten. Betrachten wir z. B. die Such- und Ersetz-Operation. Vor jeder Operation wird die Textdatei geöffnet, d. h. vom Textprozessor wird der nötige Dateideskriptor initialisiert (Datensatz und Tracknummer einlesen, Pufferzeiger initialisieren). Diese Aufgabe wird mit einfachen Systemaufrufen erledigt. Zunächst werden von der Datei einzelne Sätze in den Puffer eingelesen, solange der Puffer ausreicht. Diese Aufgabe wird mit vielen READ-NEXT-RECORD-Systemaufrufen erledigt.
Die Such-und-Ersetz-Operation übernimmt die Suche nach einem bestimmten String vom Anfang der Datei bis zum Ende. Die Datei wird dabei abschnittweise in den Puffer geladen. Die eigentliche Suchoperation wird im Puffer durchgeführt. Nach dem eventuellen Ersetzen des Strings kann die Datei aktualisiert werden. Bei größeren Dateien wird im Puffer nur ein Ausschnitt (Fenster) der gesamten Datei bearbeitet. Die Bearbeitung der kompletten Datei macht das ständige, zeitraubende Laden und Speichern von Teilen der Datei notwendig.
Sogar durch eine einfache Cursorbewegung wird dieser Mechanismus in Kraft gesetzt und führt zu spürbaren Verzögerungen bei den besten Textprogrammen, wenn im Hintergrund der Puffer aktualisiert werden muß.
Um den Anforderungen der Anwender gerecht zu werden, wurden die Adressiermöglichkeiten der 8-Bit-Prozessoren extern durch das Bank-Switching-Verfahren (Bankumschaltverfahren) erweitert (*Bild 2*). Bei diesem Verfahren setzt sich die physische Speicheradresse aus dem vom Prozessor gelieferten Offset und die über einen Peripherie-Port vorgegebene Speicherbanknummer zusammen. Der Hauptspeicher kann die Grenze von 64 KByte überschreiten, die alternativen Speicherbänke belegen jedoch die gleichen logischen Adressen und müssen z. B. mit I/O-Befehlen über den Bankswitching-Port vorgewählt werden.

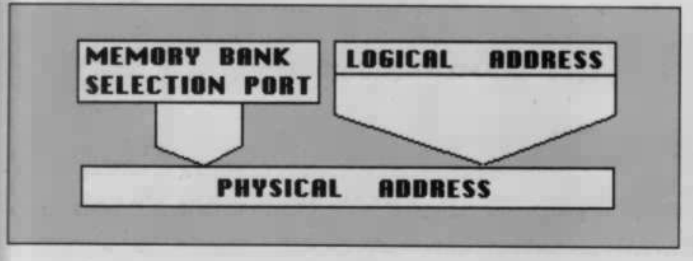


Bild 2. Die physische Adresse eines Bankswitching-Speichers wird aus der logischen Adresse und der Banknummer gebildet

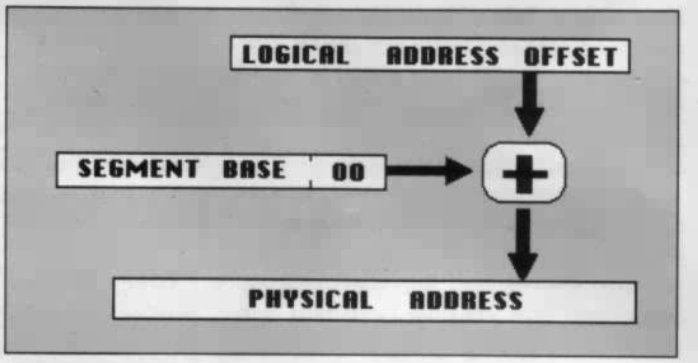


Bild 3. Berechnung der physischen Adresse beim 8086

Wegen der Verwaltungsprobleme läßt sich durch das Bankswitching der Puffer nicht unbedingt vergrößern, aber die zusätzlichen Speicherbänke lassen sich gut als RAM-Disk gebrauchen, was zur spürbaren Beschleunigung der Programmausführung und damit einer weiteren Leistungssteigerung führt.
Mit dem Bankswitching wurde die Grenze der 8-Bit-Technologie erreicht. Ohne Verbesserungen in der Prozessorarchitektur konnte man keinen wesentlichen Fortschritt weiter machen.

16-Bit-Personal-Computer der MS-DOS-Klasse

Die 16-Bit-Prozessoren bieten eine wesentlich größere Leistungsfähigkeit im Vergleich zu allen 8-Bit-Vorgängern. Die Veränderungen in der Architektur der Prozessoren erlaubten den Systementwicklern eine Erweiterung der bewährten Betriebssystemgrundlagen der 8-Bit-Computer. Außerdem kamen Impulse von der Minicomputertechnik hinzu. So entstanden Betriebssysteme mit bisher bei den Personal-Computern unbekannten Eigenschaften wie einem hierarchischen Dateisystem (Übernahme u. a. von UNIX) mit Multitaskingfähigkeiten und verfeinerten Grafikein/-ausgabemechanismen (beginnend mit dem Macintosh).
Obgleich das allgemeine Schema blieb, übertraf die Größe des internen Operationsspeichers schnell die 100-KByte-Grenze – sogar in den billigsten „Personal-Computern" – z. B. Sinclair QL. Das schaffte die Voraussetzungen für die Bearbeitung von größeren Dateien und sogar zur Unterbringung von gleichzeitig mehreren Programmen im Hauptspeicher. Mit gestiegener Leistung des Prozessors konnten nun mehrere Programme quasi gleichzeitig laufen.
Wenn mehrere lauffähige Programme gleichzeitig im Speicher untergebracht werden sollen, ist eine andere Eigenschaft notwendig: Verschiebbarkeit. Ein verschiebbares Programm ist unabhängig von der Adresse lauffähig, unter der es im Hauptspeicher geladen wird. – Es funktioniert so, wie sich der Programmierer es gewünscht hat, greift auf die richtigen Daten zu und gibt die erwarteten Ergebnisse aus. Das kann effektiv nur dann erzielt werden, wenn in den Befehlssatz des Prozessors relative Sprung- und Unterprogramm-Call-Befehle sowie relative bzw. indizierte Basisadressierung bei den Datenzugriffen aufgenommen wurden.
Um diesen Anforderungen gerecht zu werden, erlauben die meisten bekannten 16-Bit-Mikroprozessoren (Intel 8086/286, Motorola 68000/10, Zilog Z8000) die relative Adressierung nicht nur bei den Sprung- und Unterprogramm-Call-Befehlen, sondern auch in vielen Datenzugriffbefehlen. Dadurch wird sowohl das Schreiben von verschiebbaren Programmen in Assembler als auch in höheren Programmiersprachen wie C und Modula-2 vereinfacht.
Der verbreitetste 16-Bit-Mikroprozessor Intel 8086 benutzt zur Berechnung von allen Daten- und Sprungadressen die Basisadressierung: die Basisadressen von vier Segmenten (Programmcode-, Daten-, Stack- und Extended-Segment) werden in entsprechenden Registern untergebracht und bei jedem Zugriff (zuvor um vier Bit nach links verschoben) zu dem Offsetteil der Adresse addiert (*Bild 3*). Vor dem Laden eines Programms genügt es, die vier Basisregister mit den passenden Segmentbasisadressen zu laden und das Programm kann an der aktuellen Adresse gestartet werden.
Der Motorola 68000 benutzt ein anderes Verfahren: die Basisadressierung muß explizit im Programm bei allen Datenzugriffen und die relative Adressierung muß bei den Sprüngen bzw. Unterprogrammaufrufen angewendet werden. Anderenfalls ist ein Bindevorgang vor dem Laden eines Programms

notwendig, der die absoluten Adressen entsprechend der Ladeadresse aktualisiert. Die Adressierung des Motorola 68000 ist mehr physisch, die Segmentation wird nicht direkt unterstützt.
Die hardwaremäßige Realisierung der Segmentation kann mit dem Speicherverwaltungs-Schaltkreis MMU 68451 verwirklicht werden. Allerdings: die 14 im 68000-Assembler eingebauten Adressierungsarten lassen eine sehr effektive Gestaltung der Datenzugriffe sowie der Programmverzweigungen zu, so daß bei den meisten 68000-Personal-Computern auf die kostspielige MMU-Beschaltung verzichtet werden konnte.
Der Betriebssystemaufruf EXECUTE PROGRAM erledigt die Aufgabe der Initialisierung von Basisregistern bzw. die Aktualisierung der Zugriffsadressen. Zuerst aber wird durch den Allokationsalgorithmus ein genügend großer Speicherblock gefunden, der alle Segmente des Programms aufnehmen kann. Danach wird das Programm in den ausgewählten Speicherblock geladen (*Bild 4*).
Ähnlich wie bei dem CP/M-System wird vom Programmierer die Aufgabe der Verwaltung der Datenübertragung zum und vom Massenspeicher gelöst. Dabei findet zumeist das Pufferverfahren Anwendung.

Mehrere Programme gleichzeitig – Multitasking

In den Multitasking-Betriebssystemen werden mehrere Programme gleichzeitig im Hauptspeicher untergebracht und von der CPU abgearbeitet (*Bild 5*). Der Anwender kann mehrere Programme bzw. Kommandos gleichzeitig einleiten, ohne auf den Abschluß der einzelnen warten zu müssen. Somit ist das Warten auf die lange dauernde Compilation nicht mehr notwendig; zugleich kann der Programmtext weiter editiert werden.
Im Multitasking-Modus wird die Speicherverwaltung komplizierter. Seitens des Programmierers lassen sich zwei Wege der Speichergewinnung für das Programm analysieren, statisch und dynamisch.
1. Statisch.
Der vom Programm benötigte Speicherbereich wird vom Programmierer direkt im Programmtext deklariert. Zu dieser Klasse gehört auch die Deklaration von Daten in den Prozeduren in Pascal, C usw., die eine Reservierung von benötigtem Speicherbereich bewirken.
2. Dynamisch.
In Interaktion mit dem Betriebssystem erhält das Programm durch MEMORY ALLOCATE die dynamisch benutzten Speicherblöcke. Nach der Nutzung werden die nicht mehr benötigten Speicherblöcke durch MEMORY FREE zurückgegeben.
Die dynamische Verwaltung bringt besonders viele Gefahren mit sich, bietet aber die größten Vorteile: der Hauptspeicher wird besonders effektiv genutzt. Die andere Seite der Verwaltung des Datenspeichers ist die Übertragung der Daten zwischen dem Haupt- und Massenspeicher. Die dynamische Verwaltung verlangt besondere Sorgfalt bei der Gestaltung dieses Teils der Arbeit des Computers.
Die Tasks (Programme mit ihrer Softwareumgebung) werden bei den Multitasking-Betriebssystemen nicht an feste Adressen geladen, der freie Speicherbereich wird für die neuen Tasks ebenfalls dynamisch gesucht und zugewiesen. Als nützlich hat sich die logische und bei keiner Hardwareunterstützung auch physische Trennung der von einer Task verwendeten Bereiche in unabhängige Segmente erwiesen (*Bild 6*). Die gleichen Programmcodesegmente können unterschiedliche Datensegmente bearbeiten und somit durch Speicherersparnis mehrere gleichzeitig aktive Tasks bilden. Einige Mikroprozessoren unterstützen direkt die Segmentation, ohne daß eine spezielle Speicherverwaltungseinheit (MMU) benötigt wird. Bei der Segmentation werden u. a. zwei Algorithmen angewendet:
1. Segmentation mit fixierter Segmentgröße.
Diese Art der Segmentation wird selten eingesetzt. Sie hat den Nachteil, daß die Segmentgröße selten den Anforderungen entspricht und dadurch große Speicherblöcke unbenutzt bleiben.
2. Segmentation mit flexibler Größe.
Diese Art erlaubt die Anpassung der Segmentgröße an die Forderungen des Programms, ist aber aufwendiger.
Die Segmentgröße kann meistens nur bestimmte Werte annehmen (z. B. Vielfache der Basisgröße), damit ein Kompromiß zwischen der Speicherverschwendung und der Gefahr der totalen Zerstreuung der freien und benutzten Bereiche gefunden wird.
Die Gefahr der totalen Zerstreuung der Segmente nach einer gewissen Arbeitszeit des Systems gehört zu den größten Nachteilen der Speicherverwaltung mit Segmentation. Im Laufe der Zeit entstehen im Speicher Zonen von kleinen Segmenten, die immer seltener den Forderungen der Programme genügen und als Müll bezeichnet werden. Der Müll muß von Zeit zu Zeit beseitigt werden, indem die benutzten Segmente aneinandergeschoben werden. Diese Prozesse verringern die Leistung des Systems und bringen die Gefahr des Ausfalls des Systems mit sich, wenn gar keine Speicherblockanforderung durch das Betriebssystem positiv beantwortet werden kann.
Die Größe des lokalen Speichers einer Task ist bei den nichtvirtuellen Systemen durch die Größe des physischen Speichers beschränkt und, bei den Intel-8086- und Zilog-Z8000-Systemen, zusätzlich noch durch die maximale Segmentgröße, d. h. 64 KByte. Nur die Motorola-68000-Systeme kennen die 2. Einschränkung nicht, der gesamte Speicher ist linear in voller Breite des Adreßbusses adressierbar.
Ohne MMU-Beschaltung (*Bild 7*) lassen sich keine physisch funktionierenden

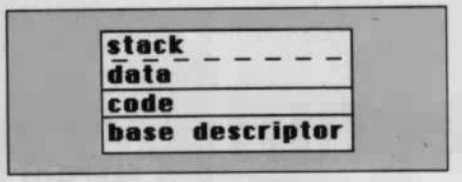


Bild 4. Programmladeblock in der MS-DOS-Umgebung

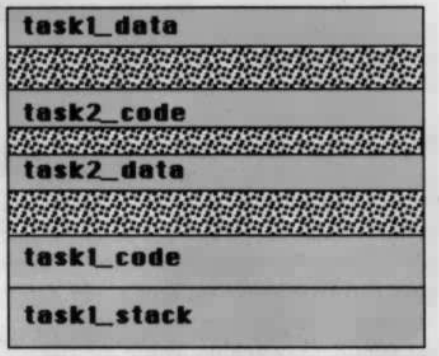


Bild 5. Segmentation in Aktion. Mehrere Programme mit ihren Daten und Stacksegmenten können im Speicher an unterschiedlichen Adressen untergebracht werden, während der Arbeit schrumpfen oder wachsen

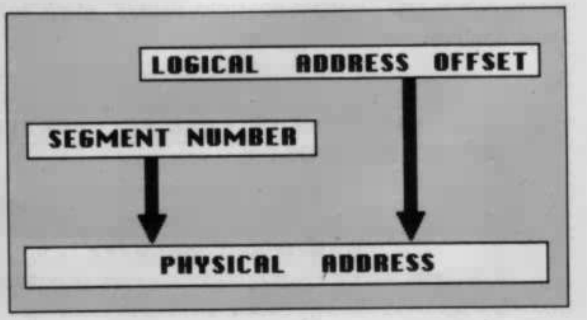


Bild 6. Berechnung der physischen Adresse in segmentierten Speichersystemen ohne spezielle Hardware-Unterstützung

SHAMROCKs SOFTWARE-Menü für PCs ...

GRAFIK

EAGLE Grafik-Editor zur manuellen Leiterbahn-Entflechtung, auch für Multilayer geeignet. Vor- und Nachbearbeitung von Autorouter-III-Dateien möglich, Ausgabe auf Drucker + Plotter; EGA + Farbmonitor erforderlich ..... **764 DM**

Autorouter III Automatische Entflechtung ein- oder zweiseitiger Platinen, Ausgabe auf Drucker + Plotter, kompatibel mit Shamrock-CAD und EAGLE, für CGA/EGA/VGA/Hercules ..... **764 DM**

Shamrock-CAD Ideal zum Schaltbilder-Zeichnen, umfangreiche Bauelemente-Bibliothek (u. a. mit allen TTL-Symbolen); kompatibel mit Autorouter III, monochrome Darstellung, Ausgabe auf Drucker + Plotter; für CGA/EGA/Hercules (bitte angeben) **495 DM**

InkPen Universelles, vektororientiertes Zeichenprogramm für EGA (16 Farben) und CGA. Symbol-Bibliotheken und selbstablaufende „Dia-Shows" programmierbar; Ausdruck auf Matrix-Printer **198 DM**

Picpaint Einfaches, pixelorientiertes Zeichenprogramm, ideal für Präsentations-Grafik, für EGA (16 Farben) und CGA **98 DM**

PC-DIA Anzeige und Druck von x-y-Diagrammen aus Meßwerte-Dateien; für CGA, EGA (in Farbe) und Hercules ... **198 DM**

KOMMUNIKATION

PCMAIL Professionelles Mailbox-System für Telefonmodem oder Datex-P20H, frei installierbare Haupt- und Untermenüs, vielfältig konfigurierbar ..... **485 DM**

TELESERVICE Fernwartungs-Programm, erlaubt Fernbedienen aller MS-DOS-Kommandos und von Anwenderprogrammen; für Akustikkoppler und Modems geeignet **485 DM**

PCTERM Terminal-Programm für PCs, ATs, PS/2, bis 9600 Bd, Dateien senden/empfangen, DIN-/ASCII-/IBM-Code, XON/XOFF ..... **98 DM**

NET.24 Low-Cost-Netzwerk für Bürokommunikation und Datei-Transfer für bis zu 26 PCs, ATs, Siemens-PCD1, Atari-ST. Starter-Kit für 2 PCs inkl. Server **303.98 DM**

PROGRAMMIER-TOOLS

EDI Superschneller Bildschirm-Editor, nur 7 KByte groß, zwei Texte mit je 64 KByte gleichzeitig bearbeitbar, alle Funktionen wie Suchen/Ersetzen, Blöcke kopieren/verschieben/laden/speichern ... **98 DM**

STRUKTOR Erstellt Struktogramme aus Pascal-ähnlichem Quelltext ..... **398 DM**

V.24-OEM Treiberprogramm für bis zu vier serielle Schnittstellen, 110 bis 38400 Bd, mit Beispielprogrammen in Assembler, C, Pascal, Basic ..... **198 DM**

PC-COMM INT-14H-Puffer für COM1 oder COM2. Erlaubt den Empfang bis 9600 Bd ohne Zeichenverlust über den AUX-Kanal von MS-DOS, mit Beispielprogrammen in Assembler und Pascal ..... **98 DM**

RS2FILE Empfängt Daten über die serielle Schnittstelle und schreibt diese in eine Datei; läuft vollständig im Hintergrund, der PC kann weiterbenutzt werden **198 DM**

NET.24-OEM Treiberprogramm zum Einbinden von NET.24 in eigene Pascal-, Basic- oder C-Programme, mit Beispielen in Assembler, C, Pascal, Basic; inklusive zwei Netzwerk-Adapter ..... **198 DM** (5 m Koax-Kabel hierzu: 7.98 DM)

Utility-Disk Drucker-Umleitung von LPT1 auf LPT2/3, COM1/2, in Datei oder auf Bildschirm; EGA-Hardcopy; Zeichenkonvertierung; schnelles Diskcopy u. v. a. .... **98 DM**

A51/MSDOS 8051/8052-Crossassembler für PCs, liefert Hex- oder Binärdateien, kompatibel zum Intel-ASM51 ..... **485 DM**

AUTO-65 Erzeugt 6502-Objektcode aus Befehlen der Siemens-NC-Sprache STEP5, mischbar mit Assembler-Quellcode **198 DM**

DATENVERWALTUNG

BRAIN Literatur-Datenbank; vielfältige Ausgabeoptionen, sehr schnelle Suche, komfortable Menüführung ..... **425 DM**

RDB Universelle Datenbank, z. B. für Adressenverwaltung; bis zu 30000 Einträge mit je bis zu 16 Feldern pro Datei **485 DM**

RECHNUNG Programm zum Schreiben von Rechnungen, dateikompatibel mit RDB **198 DM**

shamrock software

Wollen Sie mehr wissen? Bitte fordern Sie unseren kostenlosen Gesamtkatalog an – Anruf genügt.

SHAMROCK SOFTWARE Vertrieb GmbH

Karlstraße 35, 8000 München 2, Telefon 0 89/59 54 68 + 69

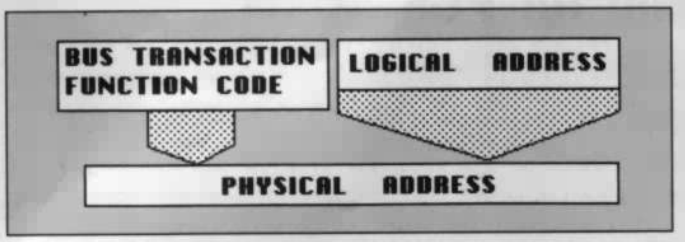


Bild 7. Sogar ohne MMU-Beschaltung läßt sich mit dem Prozessor 68000 eine Speicherverwaltung mit Segmentation realisieren

Schutzmechanismen der Speicherzugriffe einbauen. Diese Tatsache ist von besonderer Bedeutung bei den Multitaskingsystemen, wenn die Möglichkeit des unerlaubten Zugriffes und gegenseitiger Beeinflussung der Daten der einzelnen Tasks nie auszuschließen ist. Sogar mit der besten Programmierphilosophie (modulare Programmiersprachen – Modula-2 und ADA) lassen sich nicht alle Konfliktsituationen im Vorfeld, d. h. während der lexikalischen Überprüfung der Programmtexte erkennen. Ähnlich sieht die Situation bei den Systemen mit großem Massenspeicher (Festplatten) und Swappingmechanismus aus.

Virtuelle Speichertechnik

Viele Gefahren, die bei Multitaskingsystemen auf der Basis der direkten Umwandlung von logischen in physische Adressen auftreten, werden durch die virtuelle Speichertechnik beseitigt. Die virtuelle Speichertechnik trennt die logische Umgebung des Programmes (Daten, die im Programmtext deklariert und angesprochen werden) von ihrer physischen Wiederspiegelung im Operationsspeicher. Die logischen Adressen (auch in der Assemblerprogrammierung) werden durch die Speicherverwaltungslogik (MMU) der Hardware in die physischen Adressen in Abhängigkeit von der jeweiligen Systemlaufsituation umgewandelt (*Bild 8*). Der wichtigste Vorteil der virtuellen Speichertechnik liegt in der Tatsache, daß die Größe des Programmes sowie der bearbeiteten Datenbereiche weder durch die Breite der Offsetadresse noch durch die Größe des physischen Speichers beschränkt ist; es kann diese um das Vielfache überschreiten. Dabei werden keine Overlays benötigt, die Verwaltungsfunktion wird völlig vom Betriebssystem übernommen. Der Anwendungsprogrammierer ist von den meisten Speicherverwaltungsfunktionen entlastet, die auch auf der Ebene der höheren Programmiersprache sonst einzubauen sind.

Die MMU übernimmt auch die Aufgabe der Überprüfung der Zugriffsrechte und der Behandlung von illegalen Operationen. Die MMU enthält Deskriptoren mit Basisadresse, Größe und Zugriff-Flags von einzelnen Speichersegmenten bzw. -seiten (Pages).

Das Betriebssystem prüft bei jedem Zugriff, ob die angesprochene Verwaltungseinheit im Operationsspeicher anwesend ist und ob die zugreifende Task im Besitz der Zugriffsbefugnisse ist. Falls sich das angesprochene Segment (bzw. Page) im Massenspeicher und nicht im Hauptspeicher befindet, wird der zugreifende Befehl bei einer virtuellen CPU angehalten und eine Betriebssystemroutine versucht im Operationsspeicher einen ausreichend großen, freien Speicherblock zu finden. Wird ein solcher Block gefunden, kann das angesprochene Segment vom Massenspeicher in den Hauptspeicher geholt und der unterbrochene Befehl abgeschlossen werden. Sonst wird ein Replacement-Algorithmus in Gang gesetzt, dessen Aufgabe es ist, das am seltensten benutzte Segment (bzw. Page) – auch einer anderer Task – aus dem Hauptspeicher zu verdrängen (auf den Massenspeicher auszuladen), um Platz für das benötigte Segment zu machen. Zur Realisierung von virtuellen Speichersystemen wird außer der MMU auch eine virtuelle CPU benötigt (*Bild 9*): die meisten Hersteller bieten virtuelle Varianten der bekanntesten Prozessoren: Motorola 68010, Intel 80386.

Auf der Betriebssystemebene lassen sich unter Ausnutzung der MMU-Fähigkeiten weitgehende Schutzmechanismen einbauen, die eine zuverlässige Arbeit des Systems auch im Multitaskingmodus gewährleisten, unabhängig von der Erfahrung der Anwender.

Für den Einsatz in der Mikrocomputertechnik sind zwei Speicherverwaltungs-Algorithmen als bedeutend anzusehen:

1. Segmentation.

Diese Methode findet auch Anwendung in den virtuellen Speichersystemen. Sie wird erweitert um die Schutzmechanismen und natürlich um die Möglichkeit der Fortsetzung der Programmarbeit, wenn ein Zugriffsfehler auftritt (z. B. wird beim Zugriff auf ein Segment, das sich im Massenspeicher befindet, das angeforderte Segment in den Hauptspeicher geladen und die Arbeit fortgesetzt). Die von der MMU ausgeführte Umsetzung der logischen in die physische Adresse bei jedem Befehlszyklus macht alle Programme verschiebbar und hebt die Einschränkung der Nutzung der relativen Sprung- sowie Datenzugriffsbefehle auf.

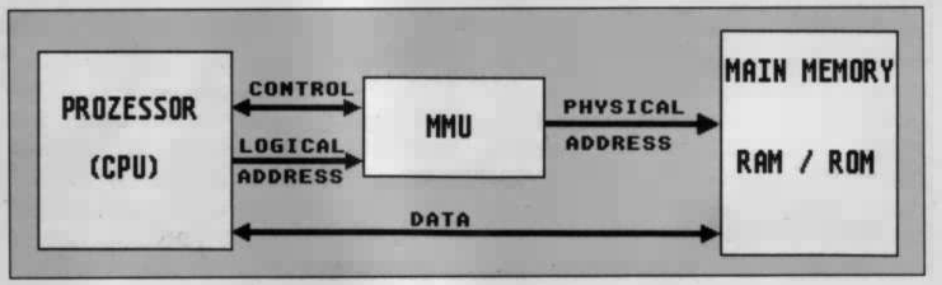


Bild 8. Schema eines typischen Mikrocomputers mit MMU

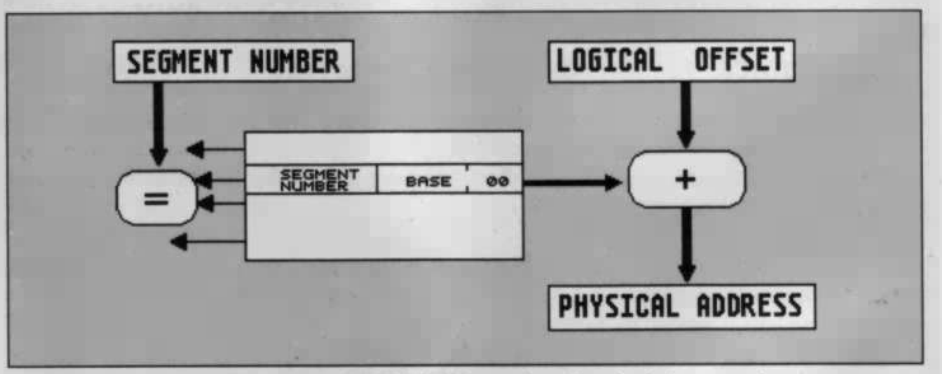


Bild 9. Motorolas Duo CPU 68010/MMU 68451 zur virtuellen Speicherverwaltung

2. Demand-Page-Algorithmus.

Bei dieser Methode wird jeder Programm- und Datenbereich in Abschnitte von gleicher Größe eingeteilt, die als Pages (Seiten) bezeichnet werden (*Bild 10*). Die Größe einer Page variiert in den meisten Systemen von 0,5 bis 4 KByte. Eine Speicherseite bildet für den Speichermanager die kleinste Verwaltungseinheit und die Übertragung zwischen dem Haupt- und Massenspeicher (Festplatte erforderlich) erfolgt je-

Damit Sie Ihr Computer noch besser versteht:

Franzis-Fachbücher

Fachlexikon der Computer

Von Herwig Feichtinger.
254 S., 40 Abb., geb.,
DM 38,–
ISBN 3-7723-8961-9

Dieses Lexikon zeichnet sich nicht nur durch die Erklärung des gesuchten Begriffes aus, sondern bietet Ihnen noch eine ganze Menge an zusätzlicher Information. Der fachliche Zusammenhang zwischen dem gesuchten Stichwort mit den angrenzenden Bereichen der Computertechnik wird Ihnen leicht verständlich aufgezeigt.

Assembler-Programmierung

Eine gründliche Einführung unter MS-DOS.
Von Wolfgang Link.
169 S., 29 Abb., kart.,
DM 48,–
ISBN 3-7723-8831-0

Hier führt Sie der Autor zum Einsatz der Assembler-Programme unter MS-DOS hin. Er berücksichtigt nicht nur die Beschreibung der einzelnen Assembler-Befehle, sondern auch die Eigenarten des Betriebssystems MS-DOS. Sind Sie Programmierer, der Programme für ATS schreibt, die auch auf PCs laufen sollen, dann werden Sie sich über die klare Gliederung der Befehle freuen.

Neuauflage

Computergrafik: Einführung – Algorithmen – Programmentwicklung

Ein Arbeitsbuch mit zahlreichen Programmen und Hardware-Applikationen für Plotter und Bildschirm.
Von Jürgen Plate. 2. Aufl.,
536 S., 178 Abb., 5 Tab.,
geb., DM 78,–
ISBN 3-7723-5006-2

Sie finden hier eine gründliche und praktische Anleitung, Computergrafiken zu erstellen, mit ihnen zu arbeiten und sie richtig auszuwerten. Dieser Band hebt sich von den anderen Computergrafik-Büchern dadurch ab, daß er sich nicht auf die reine Bildschirmdarstellung beschränkt, sondern zusätzlich umfassend die Ansteuerung von Plottern und Matrixdruckern behandelt.

Das große MS-DOS-Profi-Arbeitsbuch

MS-DOS intern – für den Profi in allen Versionen aufbereitet.
Mit PC-DOS 3.3.
Von Dieter Smode.
2. Aufl., 448 S., 71 Abb.,
geb., DM 68,–
ISBN 3-7723-8682-2

In diesem Buch finden Sie alle Versionen bis einschließlich PC-DOS 3.3 berücksichtigt. Das Haupt-Augenmerk liegt nicht auf der Mensch/MS-DOS-Schnittstelle, sondern im Bereich Anwenderprogramm/MS-DOS.

dBase für Profis

Ausgefeilte Programmier-Tricks für Insider mit dBase II, III und IIIplus.
Von Alexander Janson.
331 S., 89 Abb., geb.,
DM 68,–
ISBN 3-7723-8871-X

Dieses Buch ist randvoll mit wertvollen Hinweisen und Ratschlägen, die Sie sofort in die Praxis umsetzen können. Es ist klar und übersichtlich in Themenbereiche gegliedert.

Pascal: Einführung – Programmentwicklung – Strukturen

Ein Arbeitsbuch mit zahlreichen Programmen, Übungen und Aufgaben.
Von Jürgen Plate/
Paul Wittstock.
3. Aufl., 426 S., 184 Abb.,
kart., DM 58,–
ISBN 3-7723-6903-0

Dies ist eine systematische und gründliche Einführung in die ewig junge Sprache Pascal. Auch Turbo-Pascal als derzeitiger Pascal-Standard für Mikro- und Tischcomputer wurde berücksichtigt.

Bitte fordern Sie unser kostenloses Gesamtverzeichnis an.

Das Franzis-Fachbuch

Ein Qualitätsversprechen

Franzis-Verlag GmbH, Karlstraße 37–41, 8000 München 2

mc 9-88-34

mc

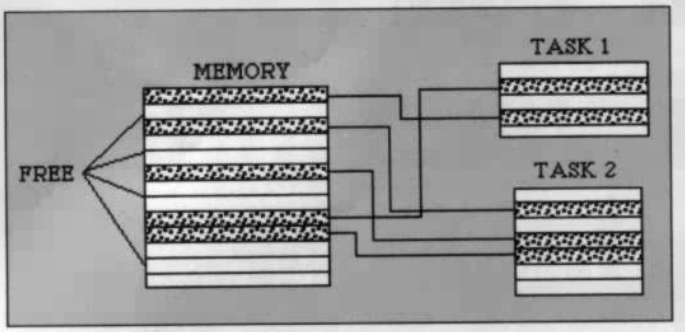


Bild 10. Virtuelle Speichertechnik mit Demand-Page-Algorithmus. Nur angeforderte Seiten belegen den Hauptspeicher

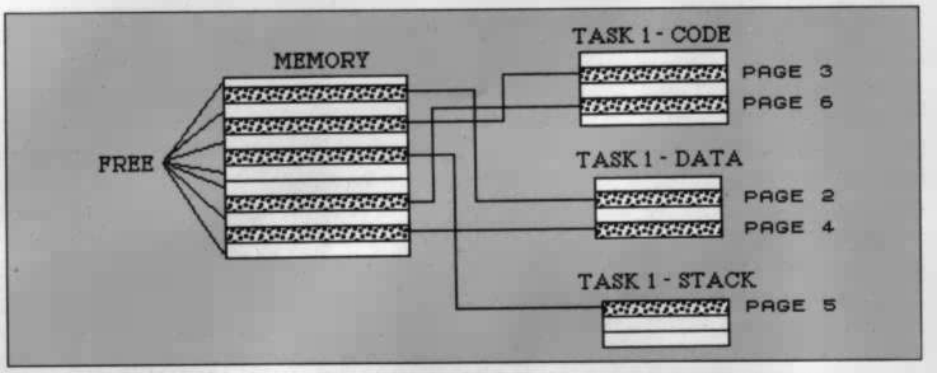


Bild 11. Virtuelle Speichertechnik mit Segmentation und Demand-Page-Algorithmus

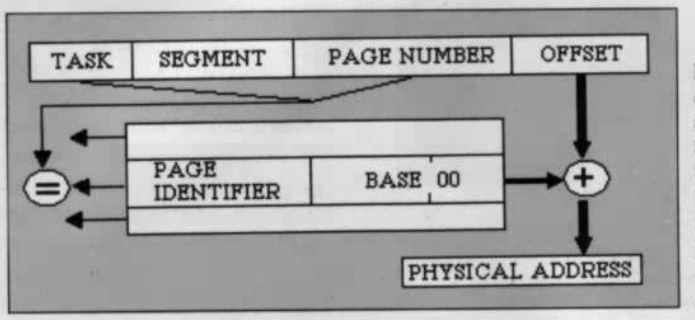


Bild 12. Berechnung der physischen Adresse in einem virtuellen Speichersystem mit Segmentierung und Demand-Page-Algorithmus. Page-Identifikatoren werden assoziativ mit der Forderung verglichen.

weils in Page-Einheiten. Im Hauptspeicher werden während eines Task-Laufes nur die angeforderten Pages untergebracht (*Bild 11*), und nicht die ganzen Segmente, was einerseits eine sparsame Nutzung des Hauptspeichers ermöglicht, von der anderen Seite her die effektive Leistung des Systems durch Einsparung der Datentransfers zwischen dem Haupt- und Massenspeicher erhöht.

Die Verbindung dieser beiden Methoden liefert die optimale Strategie für die Speicherverwaltung für Computer mit einem Hauptspeicher von einigen MByte. Der Mikroprozessor 80386 besitzt die MMU-Logik zur Implementierung einer solchen Methode. Das virtuelle Speichersystem mit Segmentation und Demand-Page-Algorithmus erlaubt sowohl die Trennung der vom Programm benutzten Adressenräume in logisch unabhängige Segmente: Programmcode-, Daten-, Stacksegment und andere, als auch eine effektive Verwaltung der Nutzung des Hauptspeichers mit Page als Grundeinheit (Bild 11).

Bei der virtuellen Technik wird die Bedeutung des Begriffes „Operationsspeicher" vom RAM-Hauptspeicher auf den Massenspeicher verlegt. Die Größe der lauffähigen Software mit den nötigen Daten wird nicht mehr durch die Größe des Hauptspeichers, sondern durch die Kapazität des Massenspeichers bestimmt. Das Spiegelbild der Tasks befindet sich im Massenspeicher, in den Hauptspeicher werden nur die gerade tatsächlich benötigten Pages geladen (*Bild 12*).

Die Speichermanager für die virtuelle Speicherverwaltung nach Demand-Page bzw. Segmentation sind wesentlich komplizierter als ihre Vorgänger, aber der Preis ist zu bezahlen, besonders wenn man sich die Profite vorstellt. Bevor die Vorteile der virtuellen Speichertechnik in den Personal-Computern voll genutzt werden können, ist eine Etablierung von einem neuen Industriestandard zu erwarten. Ist es eine Version von bereits verbreiteten Betriebssystemen (solche Lösungen sind bereits auf dem Markt oder befinden sich in der Vorbereitungsphase – Unix, MS-DOS 5.0) oder wird ein völlig neues Konzept vorangetrieben – das ist nun die Frage der nächsten Zukunft. Die virtuelle Speichertechnik wird bereits vom IBM System 2 im Betriebssystem OS/2 angeboten. Ernsthafte Konkurrenz ist aus den Reihen der Motorola-68020-Systeme zu erwarten: Apple Macintosh II gehört zu den ersten Personal-Computern mit virtueller MMU-Beschaltung. Zu lösen bleibt nur das Problem eines Betriebssystems mit Ausnutzung dieser Möglichkeiten: Unix scheint die wahrscheinlichste Alternative zu sein.

Die Zukunft

Die Entwicklung der PC-Technik hat noch lange nicht ihr Endstadium erreicht. Die neuen, qualitativen Veränderungen sind bereits in Vorbereitung. Zur Zeit setzt sich der Trend zum Einsatz von 32-Bit-Mikroprozessoren durch, wobei Intel 80386 und Motorola 68020 als Hauptfavoriten gelten. In der weiteren Entwicklung der Computerarchitektur wird die Multiprozessor-Konfiguration mehr Bedeutung bekommen. Interessant präsentieren sich die Lösungen auf der Basis von Transputer bzw. Clipper-Elementen, aber erst ein ernsthaftes Engagement von großen Firmen wird die Richtlinien bestimmen.

Auch in der Speicherorganisation sind in nächster Zukunft vielseitige Veränderungen zu erwarten. Die Größe des Hauptspeichers hat bereits die Schwelle von 1 MByte überschritten. Die Herstellung von 16-MBit-RAM-Schaltkreisen wird zur Expansion über die Grenze von 10 MByte führen. In der Massenspeicher-Technologie wurden neuartige Produkte schon angekündigt bzw. befinden sich in der intensiven Entwicklung, wodurch bestimmt die Grenze von 100 MByte auch für preiswerte Systeme überschritten wird.

Die qualitative Veränderung wird durch die Nutzung einer neuen Speicherorganisation verursacht, die im kleinen Maßstab bereits erprobt und eingesetzt wurde: der assoziative Speicher. Bei der bisher verbreiteten Organisation des Speichers werden den Dateneinheiten (Speicherzellen) eindeutige physische Adressen zugeordnet (*Bild 13*). Im Computersystem sind u. a. zwei Signalbusse implementiert, die zur Datenübertragung benötigt werden: Daten- und Adreßbus. Bei jedem Buszyklus sendet der Prozessor die Adresse der zutreffenden Speicherzelle. Diese Adresse wird vom Speichersystem decodiert, um die adressierte Speicherzelle mit einem Auswahlsignal zu aktivieren. Nach der Aktivierung der Speicherzelle können die Daten gelesen bzw. geschrieben werden (*Bild 14*).

Beim assoziativen Speicher kann der

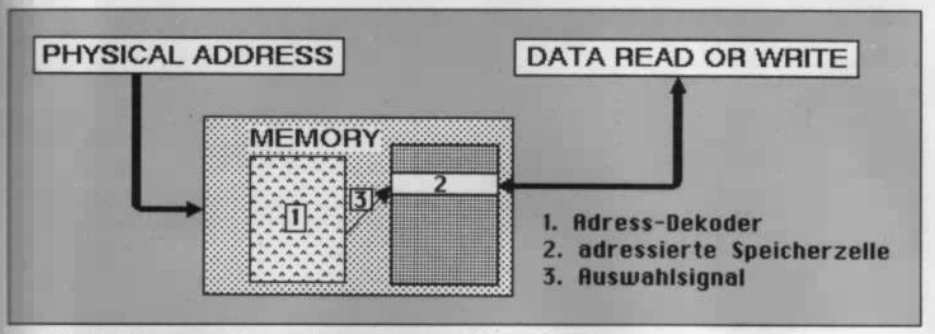


Bild 13. Speicheroperation nach dem Adressen/Daten-Prinzip

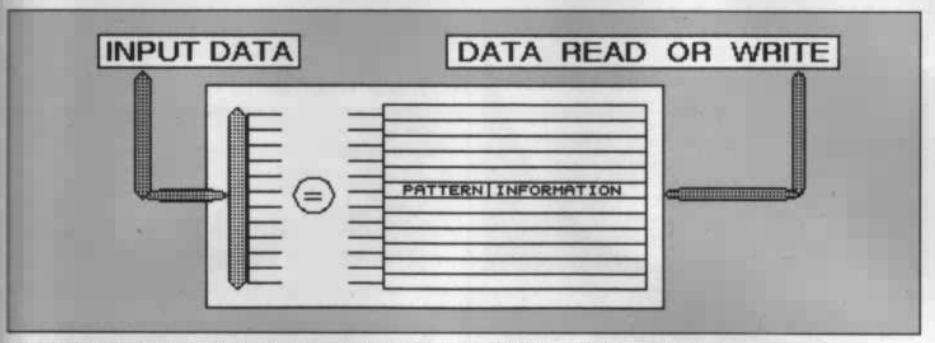


Bild 14. Speicherzugriff beim assoziativen Speicher (Daten/Daten-Prinzip)

Adreßbus praktisch eingespart werden. Der Prozessor schickt bei jedem Lese-/Schreibzyklus ein Schlüsselwort, das vom Speichersystem aufgenommen und zur Auswahl der zutreffenden Speicherzelle mit Musterworten der Zellen verglichen wird (Bild 14). Der assoziative Speicher enthält Paare (Relationen) von Dateneinheiten, die miteinander als verbunden deklariert werden. Es erinnert an den Aufbau eines Wörterbuches, in dem Wörter in zwei (oder mehreren) Sprachen miteinander verknüpft werden. Im Unterschied zum Wörterbuch aber soll im assoziativen Speicher gleichzeitig in allen Wortpaarzellen der Hardware die Suche durchgeführt werden.

Die Speicherorganisation nach dem assoziativen Prinzip findet schon heute Anwendung als Cachespeicher in hochentwickelten 16/32-Bit-Mikroprozessoren. Ähnlich werden die Deskriptoren der Verwaltungseinheiten in den MMU-Schaltkreisen organisiert. Der assoziative Speicher im größeren Maßstab wird am schnellsten in solchen Anwendungen wie Datenbanken und künstliche Intelligenz eingesetzt.

Ein wichtiges Merkmal der neuen Computerarchitekturen ist eine Orientierung auf Multiprozessorlösungen. Dieses hat auch eine Auswirkung auf die Speichersystem-Architektur. Eine Multiprozessorarchitektur setzt eine dezentrale Organisation des Speichers voraus. Zudem kann davon ausgegangen werden, daß außer den lokalen Speicherbereichen für einzelne Prozessoren im Computersystem ein globaler Massenspeicher für die größeren Datenmengen benötigt wird. Dieser Aufbau führt zu der hierarchischen Struktur des Speichers, in der die üblichen Probleme mit der Optimierung der Datentransfers auftreten. Aus diesen Gründen wird der Einsatz effektiver Verwaltungsstrategien logisch sein, wobei das entsprechende Konzept Auswirkungen sowohl auf die Hardware als auch auf die Algorithmen haben muß.

Für Anwender und Programmierer solcher Computer werden diese Mechanismen bestimmt hinter dem Vorhang des Betriebssystems verborgen sein. Besonders für die Programmierung in den höheren Sprachen (auch in den logischen Sprachen Lisp und Prolog) werden die Einzelheiten der Datenspeicherung transparent gelöst. Allerdings werden viele sicherlich die Eingriffe in die niedrigeren Ebenen der Speicherverwaltung übernehmen.

Der Spielraum für Freaks wird mit großer Wahrscheinlichkeit nicht geringer werden.

* Kurz notiert *

IBM und X/Open

Auch IBM ist jetzt der 1984 gegründeten X/Open-Group beigetreten. Damit konnte nach der Gründung der OSF (Open Software Foundation), an der IBM am 17. Mai dieses Jahres beteiligt war, nicht gerechnet werden.

Mailbox des Jahres

Der Deutsche Dachverband für Computeranwendungen e.V (DEHOCA) veranstaltet am 17. und 18. 9. 1988 in Wiesbaden den Deutschen Sysop Kongreß. Neben Referaten (Postmodem, C 64 mit 9600 Baud, ISDN) und Diskussionsrunden wird unter den anwesenden System-Betreuern die Mailbox des Jahres gewählt.

Hitachi kommt

Hitachi hat in Deutschland einen Geschäftsbereich „Bürotechnologie" etabliert. Erstmals unter eigenem Namen werden ab Oktober zwei Fernkopierer und zwei Laptop-Computer angeboten.

Grafik für Profis

In deutscher Sprache will Markt&Technik als Exklusiv-Distributor die Programme PC-Paintbrush Plus, Paintbrush Presentation Maker und Publisher's Paintbrush (DTP) von ZSoft auf den Markt bringen.

Information Management

Apollo Domain hat in München eine Information Systems Division gegründet, die Pionierarbeit im Bereich Information Management leisten soll. Neben den existierenden Konzepten des virtuellen Speicherzugriffs in heterogenen Netzen will man neue Wege und Kooperationen für branchenspezifische Applikationen suchen.

Apple legt zu

Apple erzielte im dritten Quartal (1. Juli '88) weltweit einen Umsatz von 993,1 Millionen Dollar. Das sind 56 Prozent mehr als im Vergleichszeitraum. Der Gewinn stieg um 71 Prozent auf 91,3 Millionen Dollar. Die Umsätze und Erträge von Apple Deutschland sollen noch stärker angestiegen sein.

Citizen verkauft 250 000 Drucker

Mit etwa 92,4 Millionen DM im Geschäftsjahr 1987 machte Citizen 40 Prozent mehr Umsatz als im Vorjahr und ist damit der viertgrößte Druckerhersteller Europas.

mc-QUICKIE

mc-quickies sind aktuelle Produktanzeigen, mit denen Firmen ihre Produkte vorstellen. Verantwortlich für den Inhalt sind die Inserenten.

300/1200/2400 Baud-Modem V.21/V.22/V.22bis

Tischgerät **698,–**
PC-Steckkarte **648,–**
Akustikcups **198,–**

Modemkarte PC 1200 298,–
Kurze PC-Karte, V21/22, 300/1200 Bd vollduplex,
KS 1200, extern, sonst wie PC 1200 ..... **368,–**
Pocketmodem 1200 ..... **398,–**
Alle Modems Hayes-komp. und **ohne FTZ-Nr. Anschluß an das Postnetz nicht erlaubt.**

AD/DA-Wandler 269,–
Auflösung 12 Bit, 16 AD-, 1 DA-Kanal

VIA 8255 179,–
I/O-Karte, 48 Leitungen, 16 LED, Timer/Counter

bsb
Hohenzollernring 74, 5000 Köln 1
Telefon **02 21-13 14 41**
Datentechnik GmbH

Suchen
Sie ein universelles
Programmiersystem mit
Designsoftware für all seine
programmierbaren Bausteine?

mc-AllProg plus

für GALs, Single-Chip-Prozessoren,
EPROMs, PALs, PROMs, EEPROMs,...
in verschiedenen Bauformen erhältlich bei:

H. Haase & M. Menrad GbR
Computerperipherie
Postfach 2647
3300 Braunschweig
Telefon 0531/400310

TRIO

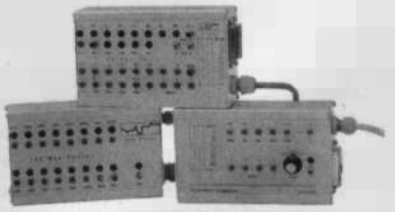
Fakturierung, Mahnwesen und Buchhaltung
Das komplette Paket für Selbständige und Kleinunternehmen:
- 3 Funktionen in einem Programm, auf einer Disk
- Einfache und sichere Benutzerführung, ausführliche Bedienungsanleitung (130 Seiten)
- **Interface für DATEV, dBASE usw.**
- Artikelverwaltung, Kundenverwaltung, Serienbriefe, Umsatzstatistik, Rechnungsbuch, Creditoren, Debitoren, Kontenblätter, Betriebsübersicht, Journal, G&V, Bilanz
- Max. 32 000 Kunden, Artikel und Buchungen
- **14 Tage frei zur Ansicht**

Und so urteilt die Presse:
„Mit sehr viel Liebe zum Detail schrieben die Programmierer ... ein Programm, das sich den unterschiedlichsten Anforderungen anpassen läßt ... Das ausgezeichnete Handbuch umfaßt mehr als 130 Seiten. TRIO ist eine Arbeitserleichterung für alle kleineren Unternehmen, die über keinen eigenen Buchhalter verfügen." PC Welt 7/88

DM 780.– bis **980.–** für XT/AT/PC

Informationen bei:

Röntgen Software
Oltmannsstraße 34, 7800 Freiburg, Tel. 07 61/40 87 40



Testgeräte für Rechnerschnittstellen

Statisches Testgerät für den IEC-Bus. Alle Funktionen als Controller, Listener und Talker möglich. Mit Anschlußkabel für IEEE-488-Bus ..... DM 513.–

Testgerät für RS-232-Schnittstellen. Alle Schnittstellenparameter über Schalter einstellbar. 16 verschiedene Übertragungsgeschwindigkeiten zwischen 50 und 19 200 Baud. Anschluß über Buchse und Stecker. Pin 2 und Pin 3 durch Schalter vertauschbar, kein Kabellöten zum Testen erforderlich ..... DM 558.60

Textgenerator zur Untersuchung an RS-232 und 20-mA. Gibt pausenlos 1 von 4 wählbaren Texten aus. Alle ASCII-Zeichen durch Anwender frei programmierbar. Speichergröße 4096 Bytes. Sonst wie RS-232-Testgerät ..... DM 478.80

Alle Testgeräte mit ausführlichem Handbuch. Solide mechanische Verarbeitung. Abgeschirmtes Metallgehäuse. Eingebautes 220-V-Netzteil.

Fordern Sie die kostenlosen Vorabinformationen an.

PETER BLANCKE Elektrotechnisches Entwicklungsbüro
Egerring 5, 6418 Hünfeld ob der Haune, Tel. 0 66 52/5 05-0

RAMs EPROMs EPROMs RAMs

Dyn. RAMs		
4164-12 (120 ns)	SAMSUNG	Auf Anfrage
41256-15 (150 ns)	NEC	Auf Anfrage
41256-12 (120 ns)	TOSHIBA	Auf Anfrage
41256-10 (100 ns)	FUJITSU	Auf Anfrage
41256-07 (70 ns)	NMB	Auf Anfrage
511000-85 (85 ns)	SIEMENS	Auf Anfrage
Stat. RAMs		
µPD 4016 C3 (150 ns)	NEC	DM 4,23
TC 55257 P10 (100ns)	TOSHIBA	DM 26,79
EPROMs		
2764-25	TEXAS INSTR.	DM 9,46
27128-20	HITACHI	DM 13,57
27256-25	INTEL	DM 13,68
CMOS-Mikroprozessoren		
V20 (8 MHz)		DM 14,08
V30 (8 MHz)		DM 19,29
Numerische Coprozessoren		
8087 - 80287 - 80387		Auf Anfrage
Festplatten-Laufwerke von Seagate		
ST 225 (21,4 MB, 65 ms)		DM 439,–
ST 238 (32,7 MB, 65 ms)		DM 464,–
ST 251 (42,8 MB, 40 ms),		DM 718,–
ST 251/1 (42,8 MB, 28 ms)		DM 952,–
ST 125 (21,4 MB, 35 ms)		DM 581,–
ST 4096 (80 MB, 28 ms)		DM 1498,–

Angebot freibleibend!

MEMORY ELECTRONICS
Inh. Ursula Nohe, Dechsendorfer Straße 10,
8522 Herzogenaurach, Telefon 0 91 32/6 11 61

ELINK-03 Datenfernübertragung ohne Probleme

- 300 BAUD vollduplex
- 1200 BAUD halbduplex
- 1200/75 BAUD vollduplex
- 75/1200 BAUD vollduplex
- ELINK-03 wählt vollautomatisch und nimmt Anrufe entgegen (MAILBOX)
- SMARTMODEM kompatibel: arbeitet mit Standardsoftware wie OPEN-ACCESS, SYMPHONY, KERMIT, CROSSTALK usw.
- Eingebaute Intelligenz erledigt Datentransfer mit jedem Computer oder Terminal über V-24-Schnittstelle, natürlich mit automatischer Baudratenerkennung
- Einfache Programmierung für Spezialaufgaben durch den Anwender
- Betrieb an Nebenstellenanlagen möglich
- Postzulassung mit FTZ-Nummer
- günstige Mengenstaffeln
- komfortables MAILBOX-Programm ebenfalls erhältlich

MICOM Computer- und Informationssysteme Entwicklungs- und Vertriebsges. mbH
Wüstenhofer Str. 6 · 5600 Wuppertal 1 · Tel. 02 02/44 34 01

OKI LASER- und MATRIXDRUCKER:
OKI LASERLINE 6 HP-Laserjet plus kompatibel . **3.495,–**
Alles über die neuen hervorragenden 24 Nadel OKI Drucker finden Sie in unserem Katalog!

PANASONIC MATRIXDRUCKER:
KX-P 1081 - 120 Z/Sek. IBM+ASCII Zeichensatz **465,–**
KX-P 1081 **plus C64/128** Interface WW 92000 nur . **499,–**
KX-P 1083 - 240 Z/Sek. neues Modell - sofort lieferbar **695,–**
KX-P 1592 - 180 Z/Sek. Breitformat 360 mm **1145,–**
KX-P 1595 - 240 Z/Sek. Breitformat 360 mm **1345,–**
Alle Modelle mit hervorragender NLQ Schrift 18x18 Matrix, Traktor und Walze für Einzelbl. serienmäßig!
Vollautomatischer Einzelblatteinzug für KX-P 1083 .... **340,–**

NEC Matrixdrucker:
Bitte fragen Sie nach unseren sehr günstigen Preisen - Wir liefern nur Originalgeräte mit Seriennummer und 1 Jahr Garantie

KABEL + DRUCKERINTERFACES
Druckerkabel 200 cm geschirmt an PC / AT ab **25,–**
Druckerkabel 200 cm beidseitig Stecker 36 polig. **49,–**
WW Interface 92008/G für C64 / C128 ..... **115,–**
WW seriell RS232/Centronics mit Buffer 8 KB **225,–**
Apple II Grafikinterface incl. Centronics Kabel **155,–**

MONITORE (Wir liefern viele Typen)
PANASONIC ST120 12" mit BAS Eingang **125,–**
VICTOR 14" TTL mit Schwenkfuß, Flachschirm, mit Kabel
sehr gute Qualität! Amber: **295,–** Grün: **340,–**
GETRONICS FM 1400, 14" TTL weiß mit Flachschirm **295,–**

THEO WEBER ELEKTRONIK, 8700 WÜRZBURG
Eisenbahnstraße 53, Tel. 09 31-70 40 91

GETRONICS MC 5400, 14" EGA der Spitzenklasse ..**995,–**
GETRONICS Multiscan 14" H15,5-35 KHz V45-80 Hz **1250,–**
NEC MULTISYNC II - neues Modell - **1495,–**
Multisync II mit Genoa Super Ega HiRes **1995,–**
NEC MULTISYNC GS, weiß mit Graustufen **575,–**

VICTOR COMPUTER
Wir führen das kpl. Programm von VICTOR vom preiswerten PC bis zum 386. Supercomputer zu unseren bekannt günstigen Bedingungen. (Kein Versand). Besuchen Sie unsere Ausstellung!

RAFI BTX DECODER
BTX PC/AT Einbaukarte mit Software und Modemkabel . **895,–**

FESTPLATTEN / STREAMER - A-Qualität!
SEAGATE ST225 - 21 MB, 65 ms - meistverkauft! ... **439,–**
SEAGATE ST238 - 33 MB, (RLL) 65 ms **495,–**
SEAGATE ST251 - 43 MB, 40 ms. mit Ontrac Softw. . **825,–**
SEAGATE ST4096 - 80 MB, 28 MB mit Ontrac **1690,–**
RODIME RO203E - 33 MB 65 ms, volle Bauhöhe **925,–**
OMTI 5520 PC Controller einschl. Kabelsatz **145,–**
OMTI 5527 PC Controller einschl. Kabelsatz **195,–**
WD 1002A-WX1 PC Contr. mit Superbios u. Kabels. . **140,–**
OMTI 8120 - AT Controller mit Kabels., Interleave 1! ..**345,–**
WD 1003 - AT Controller für zwei HD, mit Kabelsatz ...**375,–**
WD 1003-2 - AT Controller mit Floppyport **445,–**
GOLDCARD Einsteckplatte 21 MB, mit Lapine Platte . **745,–**
GOLDCARD Einsteckfestplatte 33 MB (RLL) **845,–**
ARCHIVE FASTAPE Backupsystem 20 MB für XT/AT **1195,–**
ARCHIVE FASTAPE Backupsystem 60 MB für XT/AT **1895,–**

THEO WEBER ELEKTRONIK, 8700 WÜRZBURG
Eisenbahnstraße 53, Tel. 09 31-70 40 91

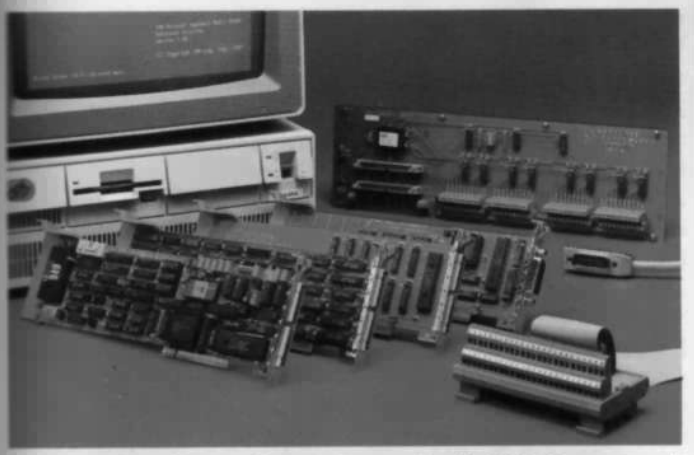
PLOTTER + SCANNER
SEKONIC SPL-410 A3 Plotter, 0,025 mm Schrittweite **1980,–**
(400 mm/Sek. 6 Farben, voll HP/GL kompatibel)
SEKONIC SPL-450, wie 410, mit 8 Farben, ... **2250,–**
SEKONIC SPL-430 A3 Friktionsplotter, HP/GL komp. **2680,–**
(8 Farben, mit Centronics und RS 232 IF).
SEKONIC SPL-1000, DIN-A0 Profiplotter .. **16.900,–**
ARISTO Digitalisiertablett A3 mit hoher
Auflösung, einschl. Fadenkreuzcursor und Datenkabel **2490,–**

Panasonic Scanner FX-RS 505, A4, max 400 P./Zoll **3150,–**
Microtek Scanner MSF 300C, A4, max 300 P/Zoll . **3950,–**
Handy Handscanner mit Software - Sonderpreis - ..**575,–**

GRAFIKKARTEN - Sonstige Karten:
EGAWONDER - neue Universalkarte für alle Monitore **450,–**
>> jetzt auch mit HiRes Modis bis 800 x 560 Punkte <<
GENOA Super EGA HiRes - 800 x 600 Punkte - **545,–**
Paketpreis Genoa Super EGA HiRes mit GEM GRAPH **645,–**
Grafikkarte **720x348** Herkules kompatibel **99,–**
Grafikkarte mit Centronicsport 720x348 - 12 MHz **195,–**
Uhrenkarte mit Kalender - mit Akkupufferung - **99,–**
Ein- Ausgabekarte 48 Bit parallel mit 3 Timer 8253 ..**295,–**
AD/DA Wandlerkarte, 16 AD und 1 DA Kanal, 0,06ms **398,–**
AT-EMS-Speichererweiterung bis 2MB (0KB best.) **345,–**

Bitte fordern Sie unseren umfangreichen Computer- und Zubehörkatalog sofort an! Wir liefern ein umfassendes Programm vieler führender Hersteller. Natürlich auch Software für viele Anwendungen. Auch Händleranfragen sind uns willkommen!

THEO WEBER ELEKTRONIK, 8700 WÜRZBURG
Eisenbahnstraße 53, Tel. 09 31-70 40 91



Das Angebot an Datenerfassungskarten für die PS/2-Computer mit Mikrokanal wächst ständig

auf 71000, die 25 µs-Version auf 45000 Abtastungen.
Bei den beiden anderen Karten handelt es sich um digitale I/O-Karten mit 24 bzw. 32 Ein-/Ausgängen.

Die gesamte Mikrokanal-Produktfamilie von National Instruments wird von der Programmentwicklungs-Software Labwindows unterstützt. Die Karten sind ab etwa 3000 DM zu bekommen.

Meilhaus
Fischerstr. 2
8039 Puchheim
☎ *0 89/80 70 81*

Für alle Color-Betriebsarten

Für den Einsatz in CAD/CAE-Applikationen, beim Desktop-Publishing und für verschiedene Grafik-Design-Aufgaben geeignet ist der Farbmonitor CM-1430 Continental. Mit einer Auflösung von 1024 x 768 Bildpunkten ist er kompatibel zum PS/2-Standard und als echtes Multi-Scan-Gerät für alle Farbbetriebsarten geeignet: VGA, MGA, EGA, PGA und für den Modus der 8514/A-Grafikkarte. Über einen als Option erhältlichen Videoausgang kann der Anwender problemlos auch Peripheriesysteme, wie beispielsweise ein Hardcopy-Gerät, ohne weitere Zusatzverdrahtung anschließen.

Böttcher Engineering
Postfach 12 27
5243 Herdorf
☎ *0 27 44/7 55*

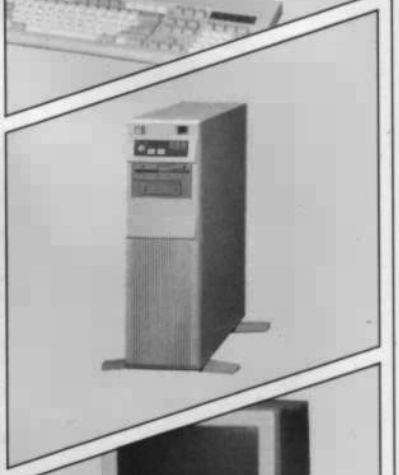
Mit einem ansprechenden Äußeren kann der Mulit-Scan-Monitor CM-1430 von Seiko Instruments aufwarten

Der Erfolg gibt uns recht – UNIC

Die benutzerorientierte Zusammensetzung von internen Bausteinen wie Zentralrecheneinheit, Speicherausbau (Haupt- und Massenspeicher), Coprozessoren, Grafikadapter, Schnittstellen und externen Bausteinen wie Tastatur, Mouse, Bildschirm, Drucker usw. ergeben das jeweils optimale System für den Benützer. Nicht nur die Hardware-Bausteine sind anpaßbar, auch mehrere Betriebssysteme (MS-DOS, XENIX) stehen zur Auswahl. Daß die verschiedenen Systemeinheiten darüber hinaus einen guten Draht zueinander haben können, wurde in der Auswahl berücksichtigt. Vom einfachen Datenaustausch zwischen zwei Systemen über die in jedem System vorhandenen Schnittstellen bis zur leistungsfähigen Vernetzung einer beliebigen Anzahl von Systemen mit Ethernet ist alles möglich.

UNIC STANDARD
Serie 60, 80:
Computer für Büro, Administration, Entwicklung und als Workstation.

UNIC	S60	S80
CPU	80286	80386
RAM Grundausst.	1 MB	1 MB
erweiterbar/CPU	1 MB	8 MB
Diskettenlaufwerk	1,2 MB	1,2 MB
Festplatte	40 MB	40 MB
Betriebssystem	MS-DOS 3.3/XENIX	



UNIC TOWER
Serie 70, 90:
Computer für Büro, Administration und Entwicklung. Als Server, Abteilungsrechner und Workstation.

UNIC	S70	S90
CPU	80286	80386
RAM Grundausst.	1 MB	1 MB
erweiterbar auf CPU	1 MB	8 MB
Diskettenlaufwerk	1,2 MB	1,2 MB
Festplatte	40 MB	40 MB
Betriebssystem	MS-DOS 3.3/XENIX	

UNIC COMPACT
Serie 20, 30, 50:
Computer für Administration, Büro und Terminal-Aufwendungen.

UNIC	S33	S50
CPU	V20	80286
RAM Grundausst.	640 KB	640 KB
erweiterbar CPU	640 KB	1 MB
Diskettenlaufwerk	360 KB	1,2 MB
Festplatte	20 MB	20 MB
Betriebssystem	MS-DOS 3.3/XENIX	

INDUTRONIC

D-8011 Poing/München, Indutronic, Dipl.-Ing. Kreiger GmbH & Co.KG, Gruberstraße 46, Telefon (08121) 701-0, Telefax (08121) 71364
autorisierte Fachhändler:
aris edv systeme GmbH, Kammergasse 5, 8050 Freising, Tel. (08161) 2807
MAXHOF-STUDIO, Schleißheimer Str. 219, 8000 München 40, Tel. (089) 3083807
Peter Hüther, Waldhäuser Str. 13, 6661 Kleinsteinhausen, Tel. (06339) 257
Delo Computertechnik, Kranenbuschstr. 28, 4600 Dortmund 15, Tel. (0231) 356511
GRÜN-DATA GmbH, Gustavstr. 16, 5510 Fürth, Tel. (0911) 746606
Dietrich's Holzbau-EDV-Programme GmbH, Lindenallee 7, 8014 Neubiberg, Tel. (089) 606078
TEAM SOFTWARE GmbH, Erbschlöer Str. 17, 5600 Wuppertal 21, Tel. (0202) 4670895
HTK-EDV-Systemberatung, Zähringer Platz 2, 7750 Konstanz, Tel. (07531) 53199
KG HANSA Hard- & Software Vertrieb, Sachsenstr. 8, 2000 Hamburg 1, Tel. (040) 233071
offsettechnik möllering datentechnik, Am Winkelsweg 95, 4600 Dortmund 50, Tel. (0231) 753821

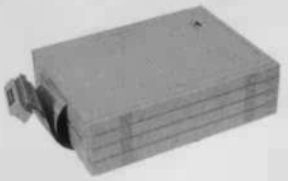
Wir bringen Sie auf die richtige Spur...

...TRACE80-ICE

In-Circuit-Emulatoren für Z80, 8085, NSC800, HD64180 oder Z180, MS-DOS – OS/2 – CP/M

- Echtzeit-Emulation bis 10 MHz
- 64(512)K Emulations-Speicher
- Beliebig viele Hardware-Haltepunkte
- Quelltext-Debugging in Assembler, C, FORTRAN, PASCAL, PL/M und PL/I
- Trace-Analysator mit 4 freiprogrammierbaren Trigger-Ebenen, 5 Zählern, 8 externe Trigger-Eingänge, 4 Trigger-Ausgänge
- Integrierte Prüf- und Simulations-Sprache, über 200 Befehle, Integriertes Handbuch
- Preis ab DM 6100,– inkl. Treiber-Software
- Entwicklungs-Paket bestehend aus Multi-Window-Editor, C-Compiler, Assembler, Linker und HLL-Debugger DM 1890,–

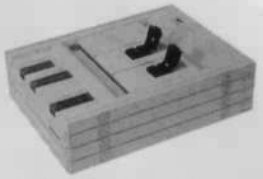
...TRACE32-ICE



Universelles, modulares Entwicklungs-System MS/DOS – OS/9 – UNIX, 68000/08/10/20/30/70 NS32008/16/32/132

- Graphikfähiges Multi-Window-Betriebssystem
- Schnittstelle zum Host über Lichtleiter, RS232 oder RS422, max. 1MBaud
- Max. 16 MByte 32-Bit-Dual-Port-Emulations-Speicher, Flagsystem, 8 Haltepunkte pro Adresse, Event-Delay-Triggerung
- Echtzeit-Debugger mit Vordergrund- und Hintergrund-Emulation
- Quelltext-Debugging in Assembler, C und PASCAL
- Frei programmierbarer Logik-State-Analysator, 120 Kanäle, 32K Tiefe, Prestore-Funktion, 64 Trigger-Ebenen, 16 externe Eingänge, 8 Trigger-Ausgänge, Time-Stamp, Performance-Analysator
- Integrierter Universalzähler, Pulsgenerator, VCO, und, und, und, ...
- Weitere Emulations-Adapter (Intel, NEC, Hitachi) in Vorbereitung

...TRACE32-XPRO



Universal-Programmier-System für EPROMs, EEPROMs, PALs, GALs, EPLDs, PROMs

- Modular aufrüstbar auf max. 120 softwaregesteuerte PIN-Treiber
- Austauschbare Sockeladapter für SMD-Bauteile
- Update-Service mit Diskette

LAUTERBACH
DATENTECHNIK
Fichtenstraße 27, 8011 Hofolding
Telefon 0 81 04 / 90 75, Telex 5 216 318 laut d
Systec '88 – Halle 11, Stand C 6
electronica '88 – Halle 13, Stand 13A04

Problemlösungen

Geschwindigkeit
Wenn Sie drucken oder zeichnen bringt ein Pufferspeicher bis zu 95% Zeitgewinn, weil der Rechner während dieser Zeiten nicht mehr blockiert ist. Für PC-Drucker bedeutet die Installation nur den Austausch des Druckerkabels:

22064i, 64K, 198 DM
22256i, 256K, 348 DM
221024i,1024K,1598 DM

V.24 mit sep. einstellb. In/Out-Parametern:
88128, 128K, 798 DM
88512, 512K, 998 DM

Datentransport
Meßdaten, Bohrpläne, Programme - jede Art von Daten kann man in einem batteriebetriebenen Buffer transportieren, der so groß wie eine Zigarrettenschachtel ist.

22032b, Centronics-Anschluß, 32K, 298 DM
88032b, RS232-Anschluß, 32K, 598 DM
88128b, RS232-Anschluß, 128K, 898 DM

Entfernung
Galvanisch getrennte Leitungstreiber werden heute als Modem-Ersatz verwendet, um Terminal oder Drucker bis zu 1200m vom Rechner entfernt zu installieren.
84boost, kompl.Psar, 1kV isoliert, 498 DM

Störsicherheit
Auch wenn Computer und Peripherie am gleichen Netz betrieben werden, können Störimpulse oder elektrostatische Entladungen zu Problemen führen. Abhilfe schaffen optische Isolatoren, die einfach in die Datenleitung geschaltet werden.
66iso, für RS422-Datenleitungen, 248 DM
88iso, für RS232-Datenleitungen, 248 DM

Terminals busförmig verdrahten
MUX-Bus: Vereinfacht die Verdrahtung erheblich, erleichtert Erweiterungen, senkt Kosten. Bis 16 Terminals und Drucker an einer vorhandenen Zweidrahtleitung, galvanisch getrennt mit Leitungstreiber bis 19200 Baud, 1000m. Info anfordern!

Umschalter
Neben mechanischen bieten wir auch vollautomatische Schalter an:

# 2i/c/0	2 PC's teilen sich einen Drucker, automatisch, inkl. aller Kabel	248 DM
# i/2c/0	Manueller Umschalter inkl. aller Kabel für Zweitdrucker	98 DM
# i/2c/64	Software-Umschalter inkl.Kabel und 64K Buffer für Zweitdrucker	498 DM
# 88128y	Intelligenter RS232-Schalter mit 4x32K	998 DM

Interfaces

von	nach
20mA	RS232
Atari 130	Centr.
C64/128	Centr,RS232
Centr.	Atari130, C128, RS232, IEEE488
IEEE488	Centr., RS232
RS232	20mA,C128,Centr,RS422,Video
RS422	RS232

Katalog anfordern!

"i"-Artikelnummer = Steckerfertig für IBM-PC und kompatible inkl. aller Kabel! Entsprechende Geräte auch für Rechner mit Centronics (36polig) Schnittstelle.

Schweiz: Weber + Co. Tel: 01 - 9302003
Österreich: Zahrer Tel: 0222 - 347671

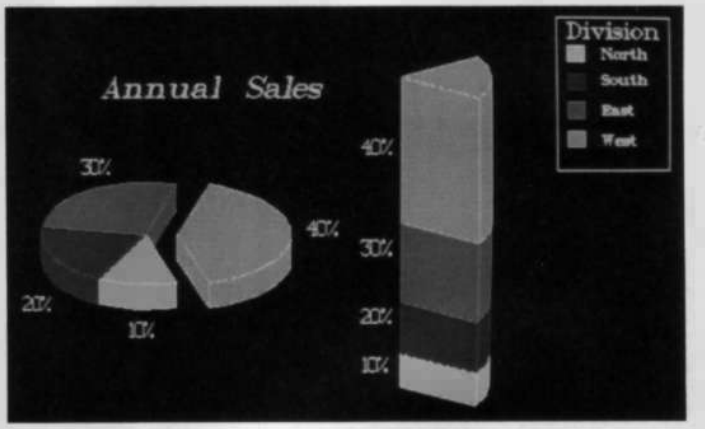
wiesemann & theis gmbh
MIKROCOMPUTERTECHNIK
Winchenbachstr. 3-5, 5600 Wuppertal 2
Telefon: 0202 / 50 50 77
Telefax: 0202 / 51 10 50
Telex: 859 16 56 wwd
Ladengeschäft: Mo-Fr. 9-17h

MARKT

SuperCalc: neue Version

Zunächst in Englisch, Anfang kommenden Jahres jedoch auch in deutscher Fassung, kommt „SuperCalc 5" heraus. Zugeschnitten auf die leistungsfähigen Prozessoren heutiger Computer, nicht zuletzt der 386er-Klasse, bietet die neue Version einige leistungsstarke Eigenschaften. Dazu gehören die Verkettung von bis zu 255 unabhängigen Spreadsheets, erweiterte Debugging-Funktionen, ausgefeilte Report-Möglichkeiten sowie die Kompatibilität zu Lotus 1-2-3.
Um den Arbeitsblättern einen hohen Aussagewert zu verleihen, kann der Anwender auch unter mehr als 100 verschiedenen zwei- und dreidimensionalen Diagrammen – einschließlich Kreis-, Balken-, Schachtel- und Flächendarstellungen – auswählen und Grafiken in präsentationsreifer Form erstellen. Zur professionellen Gestaltung stehen überdies zahlreiche Schriftarten sowie Rahmen und Schattierungen zur Verfügung.

Computer Associates
Kastanienweg 1
6108 Weiterstadt 1
☎ 0 61 50/12 00



SuperCalc 5 kennt über 100 Diagrammarten. Hier wird ein dreidimensionales Kreisdiagramm in Verbindung mit einem Stapeldiagramm gezeigt

Hochleistungs-SCSI-Chips

Anschlußkompatibel zum NMOS-SCSI-Baustein NCR5380 sind die in CMOS-Technik gefertigten, asynchronen SCSI-Chips DP8490 und DP5380.
Mit 30 ns bzw. 60 ns Quittungs-/Anforderungszeiten liegen sie in der gehobenen Leistungsklasse, gegenüber dem Typ 5380 kann der DP8490 mit dem zusätzlichen „Enhanced Mode" aufwarten, der die Durchsatzleistung steigert. Dafür sind allerdings Software-Modifikationen erforderlich, die der Anwender des DP5380 nicht vornehmen muß.

National Semiconductor
Industriestr. 10
8080 Fürstenfeldbruck
☎ 0 81 41/10 30

IEEE-488-Gerätetreiber für OS/2

Mit dem neuen Softwaretreiber NI-488 OS/2 unterstützt das IEEE-488-Interface MC-GPIB das Betriebssystem OS/2. Der neue Treiber unterstützt Multitasking-Funktionen. Die MC-GPIB mit der NI-488 OS/2-Software wird für etwa 1500 DM angeboten.
Als binärer Gerätetreiber läßt sich der Handler in OS/2 einbinden. Der Zugriff auf die IEEE-488-Treiber erfolgt über die I/O-Anweisungen der je-

Unter OS/2 erlaubt dieses Paket den Multitasking-Einsatz von IEEE-488-Systemen

weils benutzten Programmiersprache.
Da alle Funktionsaufrufe unterstützt werden, ist der Handler zu den MS-DOS- und Unix-Versionen der NI-488-Pakete voll kompatibel. Anwendungsprogramme müssen für die Umrüstung auf OS/2 deshalb nicht umgeschrieben werden. Alle Software-Bibliotheken, die für den Einsatz der Treiber erforderlich sind, liegen für C, Basic, Pascal und Fortran vor.

Meilhaus
Fischerstr. 2
8039 Puchheim
☎ 0 89/80 70 81

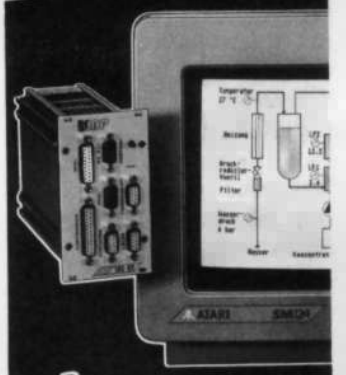
Festplatte mit maximal 170 MByte zum Nachrüsten

Schnelle Festplatten mit einer durchschnittlichen Zugriffszeit von 23 ms sind im PC-PAK, einem Hochleistungs-Winchester-Kit für PC/XT/AT-Computer, enthalten. Diese netzwerkfähigen Massenspeicher gibt es als Komplett-Konfiguration einschließlich Controller, Kabelmaterial, Partitions- und Netzwerk-Software. Im Maximalausbau liegt die Speicherkapazität bei 170 MByte. Hardware-Basis sind dabei Micropolis-Laufwerke der Serie 1370 SCSI mit ihrer beachtlichen MTBF von 30 000 Stunden.

Microscan
Postfach 60 17 05
2000 Hamburg 60
☎ 0 40/6 32 00 30

Komplett zum Einbau vorbereitet und mit Netzwerk-Software gibt es die Winchester-Kits der Serie PC-PAK

IBP
Messen, Steuern, Regeln
● Beraten
● Entwickeln
● Fertigen



z.B.:
190 ST ATARI-
kompatible Industrie-Lösung

Basierend auf dem ST-Konzept hat IBP eine eigenständige Industrie-Lösung entwickelt.

- Voll Mega-ST kompatibel
- Buserweiterungen integriert: MEGA-ST-, VME-, ECB-, SMP- und EURO-BUS (vielfältig ausbaubar).
- Watchdog/Echtzeituhr
- Steckplätze für: Anwendersoftware Coprozessor Blitter
- Extrem kompakte 19"-Kassette (3 HE/14 TE)

IBP bietet eine Fülle von Karten (AD/In-/Output) für verschiedenste Aufgaben und komplette Problemlösungen.

IBP
IBP Gerätebau GmbH
Lilienthalstraße 13
3000 Hannover 1
Telefon 0511/63 09 63/64

Krepp-Papier – adieu!

LAYO1 V3.00 ist da!
LAYO1 ist eine professionelle Leiterplatten-Layout-Software.
– sehr leistungsfähig
– sehr preisgünstig
nur DM 2850,–
+ MWSt.
(inkl. MWSt.: DM 3249,–)

LAYO1 bietet:
- 16 verfügbare Ebenen
- Auflösung 0,02 mm oder 1/1280 inch
- Zeichnungsgröße bis zu 650 x 650 mm
- 16 Standard-Lötaugen, 8 Bohrlochdurchmesser, 7 Leiterbahnbreiten von 1/80 bis 13/80 inch, SMD-fähig
- Schrittgrößen jederzeit veränderbar von 0,01 mm bis 100 mm bzw. 1/1280~1 inch
- Jederzeit umschaltbar mm ↔ inch
- Große, leicht erweiterbare Datenbank für Bauteilsymbole und Teilzeichnungen
- ASCII-Zeichensatz integriert
- Alle Korrekturmöglichkeiten wie: Kopieren, Verschieben, Drehen, Löschen
- Fenster- und Gummibandtechnik

Ausgaberoutinen für
- viele Standardstiftplotter
- Nadeldrucker
- Glaser- und Gerberfotoplotter
- Maßstab einstellbar von 0,2 ... 10:1
- Bohrplan auf Diskette, Papier oder Lochstreifen

Benötigte Hardware:
- PC-XT oder AT mit 512 kB MS-DOS 3.0 oder höher
- 2 Laufwerke oder Harddisk
- (CGA)- oder EGA-Grafikkarte

Deutsches Handbuch verfügbar

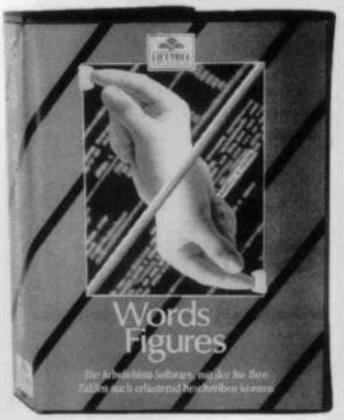
Arbeitsfähige Demodiskette (CGA) gegen V-Scheck über DM 10,–

micro-tech
Elektronik GmbH
Weltenburger Straße 6
D-8000 München 80
Telefon (0 89) 9 10 12 34
Telex 5 28 373 mtech d

Preisgünstiges „Words & Figures“

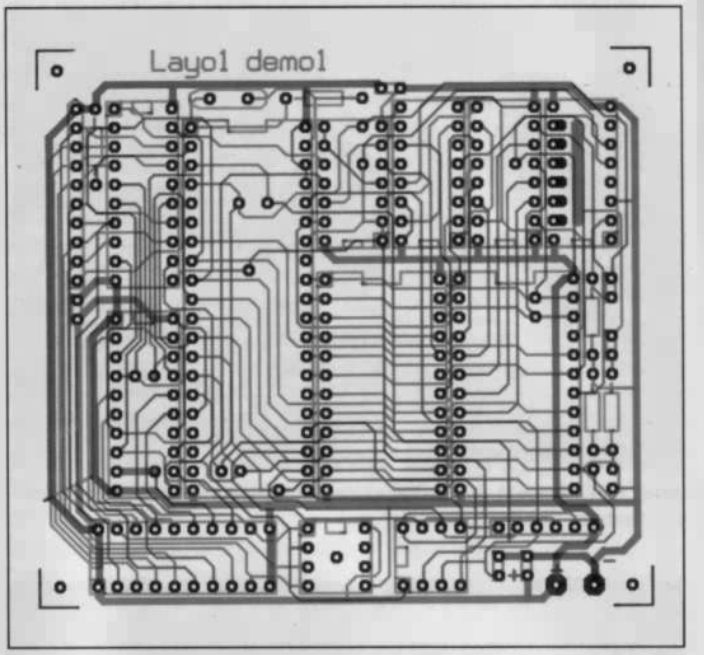
Jetzt ist es zum Preis von 299 DM zu haben, das zur Version 1A von Lotus 1-2-3 kompatible Spreadsheet-Programm “ Words & Figures“ in der deutschen Fassung. Das Programm besitzt eine integrierte Textverarbeitung, arbeitet mit maximal 9999 Zeilen und 256 Spalten, kann einen Fehlersuch-Modus während des Rechenvorgangs anbieten und erlaubt auch die Bearbeitung verborgener Rechenzellen. An den gängigen Standard angepaßt ist der Texteditor. Als Besonderheit bietet das Programm, aktive Rechenvorgänge unmittelbar in Textdokumente zu übernehmen. Die Inhalte dieser Rechenzellen bleiben auch in feststehendem Text ständig variabel und werden erst auf Befehl „eingefroren“.

Zusammen mit dem Programm kann man alle EGA- und Herkules-Grafikkarten einsetzen. Im Lieferumfang des Softwarepakets (5¼- und 3½-Zoll-Disketten) enthalten ist ein ausführliches, deutschsprachiges Handbuch.



Für 299 DM ist das Spreadsheet-Programm „Words & Figures“ in der deutschen Ausgabe erhältlich

Lifetree Promotion
Dachauer Str. 20
8066 Eschenried
☎ 0 81 31/70 01 50



„LAYO 1“ arbeitet nach der Gummibandtechnik, kann SMD-Bausteine plazieren und enthält ein komplettes Bohrprogramm

Layout-Programm – auch für SMD

Nicht nur mit normalen DIL-ICs und allen anderen handelsüblichen Bauelementen, sondern auch mit oberflächenmontierbaren Chips (SMD – Surface Mounted Devices) arbeitet das auf jedem AT-kompatiblen Rechner lauffähige Layout-Programm „LAYO 1“. Es verarbeitet bis zu 16 Entwurfsebenen mit einer maximalen Auflösung von 0,02 mm und läßt auf dem angeschlossenen Plotter das Layout mit einer maximalen Zeichnungsgröße von 65 cm × 65 cm erscheinen. Zur Verfügung stehen 16 Standard-Lötaugen, acht verschiedene Bohrloch-Durchmesser sowie sieben Leiterbahnbreiten. Die Bauteil-Symbolbibliothek ist bereits umfassend angelegt, sie kann jedoch jederzeit vom Anwender den eigenen Bedürfnissen angepaßt werden. Bedient wird das Programm durch einfache Tastenkombinationen bzw. mit der Maus – das Training sollte deshalb, so das Vertriebsunternehmen, nicht mehr als eine Stunde in Anspruch nehmen. Die Darstellung auf dem Bildschirm erfolgt im hochauflösenden EGA-Modus mit 620 × 350 Bildpunkten.

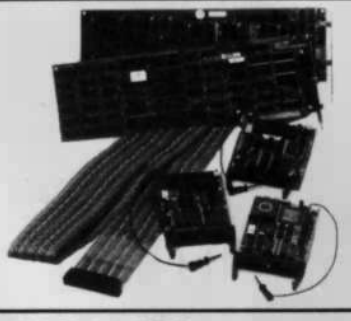
Micro-Tech
Weltenburger Str. 6
8000 München 80
☎ 0 89/9 10 12 34

iSYSTEM
Gesellschaft für Informatiksysteme mbH - Einsteinstr. 5 - 8060 Dachau
Tel. 08131/1686 - Fax. 14024 - Tx. 527559

EMUL51-PC

Die Revolution in der IC-Emulatortechnik

8051- Emulator als Steckkarte für den PC



- 16MHz in Echtzeit
- 64K Programmspeicher
- 64K Datenspeicher
- 16K*48Bit Tracespeicher
- 18 externe Signale
- 10 Breakpoints
- Realtime und Simulation
- mächtigeTracefunktionen
- Tracebedienung "on the fly"
- Windowtechnik
- Online-Help
- interaktive Menuetechnik
- Befehls-Macros
- symbol. Debugging
- Hochsprachensupport
- Assembler/Disassembler
- Performanceanalyse
- zuverlässig und bewährt
- flexible Modultechnik
- Personalitymodule für:

80(C)31	80C31-1
80(C)32	8344
80C252	80C321
8051/52/53	8751/52
80C152	80C154
80C451	80C452
80535	80552

NOHAU ELEKTRONIK AB

z.B. Emulator, 128K, 12MHZ mit POD31, komplett mit Software und Handbuch 6624,-DM

Intelligente Lösungen für Ihre Probleme

Hard- und Software zum Themenkreis „Datenerfassung" kompakt beisammen: Gratis-Buchset von Data Translation

Meßdatenerfassung und Bildverarbeitung kompakt

Nach dem bereits vor einigen Wochen von Data Translation angebotenen Buchset zum Thema „Digitale Bildverarbeitung" kommt jetzt – gleichermaßen kostenlos – ein Bücherpaket „Datenerfassung" heraus. Es umfaßt neben einer allgemeinen Produktübersicht eine 600seitige Sammlung aller verfügbaren Datenerfassungs-Platinen für die Bussysteme XT/AT, PS/2, Macintosh II, Bit-Bus, MicroVAX, Apollo, Sun, Unibus, Multibus/II, VMEbus und STDBus. Darüber hinaus sind alle Module und Anschlußleisten aufgeführt. Zusätzlich findet der Leser auf 110 Seiten spezielle Applikationen und neben den reinen Beschreibungen der Hardware-Komponenten auch die Erläuterungen zahlreicher Software-Treiber und -Komplettpakete.

Data Translation
Stuttgarter Str. 66
7120 Bietigheim-Bissingen
☎ 0 71 42/5 40 25

AT und PS/2 kommen zusammen

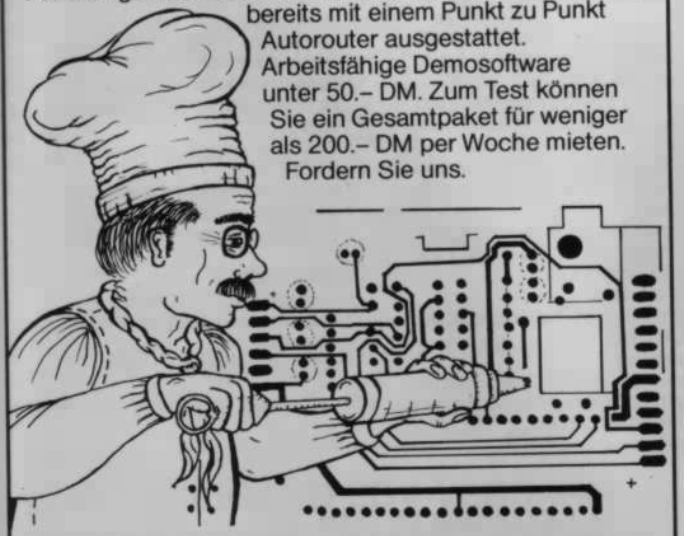
Eine Verbindung zwischen der altbewährten AT-Welt und den PS/2-Computern können die BridgeFile-Floppy-Systeme herstellen. Es gibt davon zwei Varianten: die eine mit einem 5¼-Zoll-Floppy-Laufwerk zum Anschluß an die PS/2-Modelle, eingebaut in ein Slimline-Gehäuse. Mitgeliefert wird auch die entsprechende Software sowie der zum PS/2-Modell passende Controller. Das zweite BridgeFile-Modell bedient sich eines 3½-Zoll-Laufwerks mit 720 KByte bzw. 1,44 MByte Kapazität. Auch dieses Laufwerk ist in ein Slimline-Gehäuse eingebaut und wird inklusive Software anschlußfertig für AT-Computer mit dem richtigen Controller geliefert.

Impec
Waldhörnlestr. 18
7400 Tübingen
☎ 0 70 71/7 00 20

Erstellen Sie Ihre Leiterplatten-Vorlagen noch so? Dann ist es Zeit für den DRAFTSMAN-EE™

Von der Stromlaufplanerstellung bis zum Fotoplot bietet unsere Software alles – in professioneller Qualität und 1 mil Auflösung.

Für weniger als 2500.– DM erhalten Sie das Grundpaket, bereits mit einem Punkt zu Punkt Autorouter ausgestattet. Arbeitsfähige Demosoftware unter 50.– DM. Zum Test können Sie ein Gesamtpaket für weniger als 200.– DM per Woche mieten. Fordern Sie uns.



WALTER ELECTRONIC
7901 Schnürpflingen · Lindenstr. 3
Tel. 07346/8910 · Fax 07346/8930

Rheintec

☎ (0211) 68 32 42

Logitech C7 plus Package – Mouse
mit deutschen Handbuch nur 199.-
mit englischen Handbuch nur 168.-
X5 Drei Tasten Mouse nur 69.-

Rein 300SLC 286-Laptop ++ 10MHz ++ Gas-Plasma-Bildschirm 640x400 EGA + + 20 MB Festplatte ++ 1.2MB 3.5" Floppy ++ 640KB RAM (aufrüstbar bis 2.6MB) + + zwei seriell u. ein parallel-Port ++ 6,4Kg ++ elegantes, schwarzes Gehäuse 32 x 32 x 9,4mm Superpreis auf Anfrage!!!

NEU! NEU! NEU! NEU! NEU! NEU! NEU! mit SUPERSET 30 plus (PS/2-30 kompat.) + + 8086 – 10MHz ++ 640KB RAM bestückt ++ VGA Karte ++ seriell/parallel-Port ++ 1 x 3.5" 720KB Floppy-LW ++1 x 31MB Netto (35mS!) Festplatte ++ große Tastatur ++ ohne Monitor Superpreis 3249.-

8088 XT – 10MHz Mono-System ++ 640 KB RAM bestückt ++ Herkules Karte + + seriell/parallel-Port ++ große Tastatur ++ 1 x 360KB Floppy-LW ++ 1 x 31MB Netto (35mS!) Festplatte ++ ohne Monitor Superpreis 2394.-

NEC 3.5" 720KB Floppy-LW	nur 228.-
ST251 – 1 Festplatte (28mS!)	nur 898.-
ST138R – Festplatte (35mS!)	nur 598.-
KYOCERA 20MB Festplatte (62mS!)	nur 456.-
KYOCERA 30MB RLL (62mS!)	nur 499.-

NEC P2200	NEC P6-plus	NEC P7-plus
nur 898.-	nur 1698.-	nur 2098.-

EPSON Drucker	Preis auf Anfrage!
TOSHIBA P321SL	Preis auf Anfrage!
PANASONIC Drucker	Preis auf Anfrage!
TECO VP1814 Drucker	nur 549.-
FUJITSU DL3400 Drucker	nur 1949.-

RHEINTEC 286 – 12MHz Mono-System 80286 ++ 6/12MHz ++ 512KB RAM (bis 1MB aufrüstbar 640/384KB) ++ 1 x 1.2MB 5.25" Floppy-LW ++ Herkules komp. Karte ++ seriell/parallel-Port ++ gr. deutsche Tastatur ++ ohne Monitor mit 31MB Netto (35mS!) Festplatte nur 2995.-

RHEINTEC286 – 12MHz VAG-System mit VGA-Karte 64KB (28mS!) Festplatte ohne Monitor nur 4995.-

Aufpreis 3.5" 720K Floppy nur 228.-

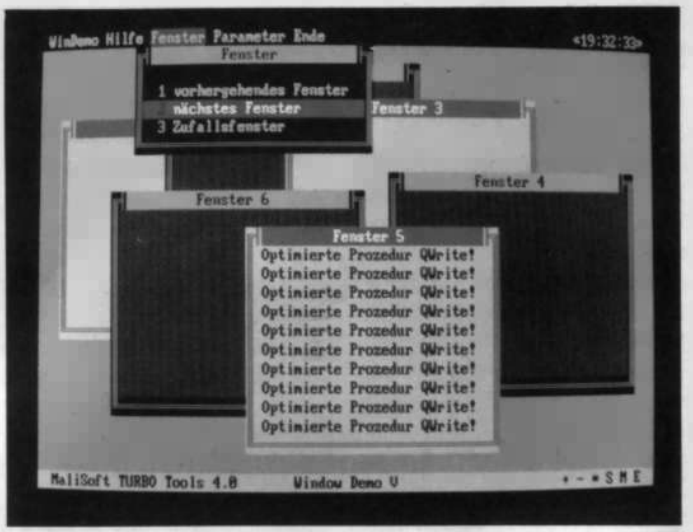
CTX Multisync (800 x 560)	nur 1099.-
HITACHI Multi560 Multisync	nur 1197.-
PANASONIC TR170 A4 17" papierweiß Monitor mit Controller	nur 4980.-
EIZO 14" Flexscan 8060S	nur 1596.-
EIZO 16" Flexscan 9070S	Preis a. A.
ADDONICS 14" Mono-Multisync	nur 598.-
NEC MultiSync GS	nur 529.-
NEC MultiSync II	nur 1598.-
NEC MultiSync Plus	nur 2299.-
NEC MultiSync XL	nur 4999.-
14" TTL Flattscreen Monitor	nur 299.-
GENOA Super EGA HiRes plus, mit VGA	nur 490.-

VGA VGA VGA VGA

ORCHID DESIGNER VGA Karten, 512KB RAM, (1024 x 768 Pixel)	nur 998.-
256KB RAM, (800 x 600 Pixel)	nur 698.-
Paradise VGA-Plus Karte	nur 798.-
Video Seven VEGA-VGA Karte	nur 798.-
ATI V.I.P. VGA Karte	nur 599.-

Rheintec Computer GmbH
Platanenstr. 11
4000 Düsseldorf 1
Telefon 02 11/68 32 42
Telefax 02 11/6 80 32 93

Öffnungszeiten:
Mo-Fr. 10.00-13.00 u. 14.00-18.30
Sa. 10.00-14.00



Komfortable Bildschirmverwaltung unter Turbo-Pascal 4.0 mit den MaliSoft-Tools

Tools für Turbo-Pascal 4.0

Mit über 300 Prozeduren und Funktionen, die hauptsächlich die Bildschirm- und Datei-Ein/Ausgabe auf IBM-XT/AT-Systemen unterstützen, können die MaliSoft-Turbo-Tools für die Version 4.0 von Turbo-Pascal aufwarten. Die Arbeitsweise dieser Hilfs- und Zusatzprogramme ist vom erfahrenen Programmierer und Turbo-Pascal-Kenner rasch erfaßbar und deshalb auch leicht in bestehende Applikationen einzubauen. Das Softwarehaus richtet sich mit diesem Produkt deshalb spezifisch an den Profi-Programmierer, der seine Software mit modernen, leistungsfähigen Benutzeroberflächen versehen will, ohne hierfür unnötige Zeit zu investieren. Dabei ist es egal, ob die Routinen in Textsysteme, Tabellenkalkulations-Programme oder integrierte Pakete mit komfortabler Bildschirmsteuerung eingebunden werden sollen.

MaliSoft
Liedolsheimer Ring 26
7500 Karlsruhe 31
☎ 07 21/78 51 79

Hochauflösender Farbdrucker

In nahezu Foto-Qualität arbeiten die Thermotransfer-Farbdrucker der VCC-Serie, erhältlich jetzt als Varianten mit 12 Punkten/mm Auflösung (bereits im Markt eingeführt sind Modelle mit 8 Punkten/mm). Dabei lassen sich aus den acht Grundfarben praktisch beliebig viele Mischfarben durch Rasterung erzeugen – als Druckmedium kann Papier oder Folie verwendet werden. Für eine Seite im DIN A4-Format braucht dabei der Typ VCC 345 rund eine Minute. Die gleiche Auflösung, jedoch das größere Papierformat DIN A3, kann demgegenüber das Modell VCC 645 bieten.
Als Option gibt es für beide Drucker spezielle Video-Schnittstellen zum unmittelbaren Anschluß an einen Farbmonitor (Hardcopy-Betrieb).

Kontron Datensysteme
Freisinger Str. 21
8057 Eching
☎ 0 81 65/70 71 39

OADC 3 in 1
The Best Solution of Image Processing

OAScan — Handy Scanner Handy Publishing Editor

OAMouse — Optical Mouse Halo Paint & Graph

OAFax — Advanced Fax Card

APPLICATIONS
- DeskTop Publishing
- Fax Transmission
- Optical Character Recognition

Fully Integrated Scan-Edit-Transmission Solutions

OAScan features:
The HPE (Handy Publishing Editor) is a bundled software providing the following advanced utilities and features:
* Move, Cut, Copy, Paste, Erase, Reverse, Save, Print, ..., etc.
* Image editing tools: Text typing, Zoom editing, Region painting, Circle and Line drawing, Eraser Selection area, Brushing, ..., etc.
* 7 file formats (Tiff, Window Paint, PaintBrush, Dr. Halo, Gem, OAFax, OCR) that can be used with other popular DTP softwares, FAX transmission, and Character Recognition.

SPECIFICATIONS
* Resolution: 400/300/200 dpi
* Scanning width: 105mm/64mm
* Scanning modes: B/W, Halftone 1-2-3
* Scanning Speed: 3ms/line
* Model: HS-3000/5000/5100/7000

OAMouse features:
* Higher resolution (up to 500 dpi)
* Maintenance free-no mechanical ball
* Fully Microsoft Mouse and Mouse System mouse compatible & switchable (by H/W)
* Provides a Turbo mode for quick movement to save valuable desk-space
* No special optical pad required
* With Halo Paint and Graph software
* Compatible with AutoCAD, MS-Windows, Dr. Halo, Lotus 1-2-3, Framework, ..., etc.

SPECIFICATIONS
* Technology: optical, microprocessor control
* Buttons: 3
* Interface: RS-232C, DB9 Female, Optional DB25 to 9 Female connector for IBM PC/XT/AT, PS/2 or compatibles

OAFax features:
* 20 mail boxes (max 16000 mails) for Broadcasting and Scheduling sending
* Foreground or background receiving and auto printing
* Supports File Transfer function to send ASCII file up to 9600bps high speed
* Text and graphic edit/convert/merge
* Links with FAX machine-works as a scanner or printer
* Option Auto power ON/OFF Box

SPECIFICATIONS
* Compatibility: CCITT Group III
* Transmission speed: 9600/7200/4800/2400 bps
* Data compression: 1D/2D

Distributor/ Importer Wanted

OADC

OA DATACOMM CO., LTD.
Head Office:
6Fl., No. 281, An-Ho Rd.,
Taipei. Taiwan, ROC
Tel: 886-2-7388094, 7386460
Tlx: 14286 OADC Fax: 886-2-7366733

USA Branch Office:
One World Trade Center
Suite 2841, New York NY 10048,
U.S.A. Tel: (212)8390902
Tlx: WUI 640134 DHI NYK
Fax: (212)4666761

mc-MINIMARKT

BIETE AN SOFTWARE

Engl. u. deutsche PUBLIC-DOMAIN-Software. 6 Katalog-Disketten gegen Einsendung eines 10-DM-Scheins. Auch auf 3,5" (720 KByte) lfbar. G. Strübbe, Wiesenstr. 5, 8543 Lienen

Ladbare Zeichensätze, mathematische & chemische & Formeln, Fonteditor für Sonderzeichen auf Bildschirm und Drucker, mit MS-Word, WordPerfect 4.2 & 5.0, für LaserJet, NEC, Hercules Plus, EPSON. Demodiskette für 5.– DM: Hans-J. Simon Verlag, Postfach 25 42, 5300 Bonn [G]

Software	
Pascal 4.0 Neu	269.–
GFA Fakt Neu	99.–
Speed Fakt Neu	99.–
Speed.Lib Neu	99.–
Witchpen Ver 1.10	89.–
Dos Merge 286 Neu	299.–
M. Gribs, Am Sportplatz 51,	
4005 Meerbusch 2,	
☎ 0 21 59/37 06	

OS/9 für MC68000/MC68010 und CP/M68K für MC68010. Preis VS. ☎ 02 14/2 73 42 (werktags ab 18 Uhr)

Super PUBLIC-DOMAIN-SOFTWARE nur ausgesuchte/getestete Disk. verk. u. tausche ich privat: ☎ 0 20 43/2 17 23, ab 17 Uhr

LOK-SOFT für MÄRKLIN DIGITAL Profi-Modellb.Steuerung f. alle Anlagen. Unterst. alle Funkt. v. MÄRKL.DIG. Indiv. Fahrprogr. für jede Lok. Abl.Steuerung d. Schrittketten. Kompl. Bedienung über Pull-Down Menüs. Info anf. Für MS-DOS. DM 298.–. Wolfgang Schapals, Franz-Wunner-Str. 24, 8948 Mindelheim, ☎ 0 82 61/96 15 [G]

■ Angst vor Computerviren??? ■
■ Virucheck meldet sofort jede ■
■ Änderung von COM und EXE ■
■ Dateien. **NUR DM 99.–** ■
■ GAP mbH, 3300 Braun- ■
■ schweig, Bolchentwete 3, ■
■ ☎ 05 31/79 77 17 [G]

Diskettenverwaltungsprogramm für IBM-PC, XT, AT 40 DM, 20 PD-Disketten 50 DM. ☎ 0 21 73/5 05 06

Programmierer, Stud. ZBW, 10 J. Programmiererf. schreibt mathematisch-wiss.-technische Programme n. Ihren Wünschen zu Festpreis. Einfach anrufen! 4700 Hamm, ☎ 0 23 81/8 86 20

Deutsche Anleitung f. PC-TOOLS deluxe DM 19.80. ☎ 0 52 41/5 67 47

Public-Domain-IBM + komp. je Disk 5 DM o. Mehrkost. Katalog frei. A. Ristau, Peetzweg 9, 3320 Salzg. 1 [G]

PROTEL-PCB mit Autorouter (original) für 1800 DM (VB) zu verkaufen. ☎ 0 69/23 37 98

FORMATKONVERTER: Schreiben und Lesen von **CPM**-Disketten auf **IBM-PC/XT/AT. DM 128.–.** Info bei: **HüWiCo,** Am Siepen 17. 4630 Bochum, ☎ (02 34) 36 12 06 [G]

LAYOUT I: Schaltpläne erstellen, >200 Symb. (erw.), 128.–. LAYOUT II: Platinen-Entwurf, 100 % mausgest., Auto-Router, Grafik-Menüs, ... 128.–. »Beide Programme für: 220.–.« Dipl.Ing. R. Groß, Oberwiesenweg 1A, 6108 Weiterstadt, ☎ 0 61 50/5 28 60

MZ-80B-FREEKS! Schnelle Graphik-Routinen + CRT-Ausdruck im Turbo-Pascal-Source-Code kostenlos gegen formatierte Diskette (CPM 2.2/Rückporto). Dr. Hardtke, Weitzmühlener Str., 2816 Kirchlinteln. Suche selbst: Tools für Turbo-C.

Restbestand ATARI-Software? Nur ORIGINAL-SW! Supergünstig, bis 50 % unter Neupreis. Liste: A6/BTX, Postfach 35 08, 6750 Kaiserslautern

Z80-fig-FORTH (CP/M), frei geg. form. 8"-/5¼"-Disk u. Rückporto. E. Ramm, Postf. 38, 2358 Kaltenkirchen, ☎ 0 41 91/16 21

PC-Freeware, 1700 Disk, 3–5 DM; ATARI-ST – PD, 400 Disk, 5–7 DM. Info frei, Kat. PC 10 DM, ST 5 DM. Johrend, Neusalzer Str. 9, 8500 Nbg.

Kassenbuchführung f. PC u. Komp. F. Kleinbetr. u. Selbständ., auch als Haushaltsbuch geeignet. Eingabe durch def. Kurzzeichen. USt-berechnung, automatisches Sortieren. DM 48 Demo. DM 10. H. Maier, Neuhäuserstr. 4, 8501 Burgthann [G]

Engl. u. deutsche PUBLIC-DOMAIN-Software. 6 Katalog-Disketten gegen Einsendung eines 10-DM-Scheins. Auch auf 3,5" (720 KByte) lfbar. G. Strübbe, Wiesenstr. 5, 4543 Lienen

Endlich ist es da – das Public Domain und Shareware-Handbuch, auf das Sie schon lange gewartet haben.
PD-Handbuch (90 Seiten) DM 5.–
oder
PD-Handbuch mit 2 Begleitdisketten
– Disk A: Program Reviews
– Disk B: Intro Disk DM 14.80
Lieferung per NN (+ DM 6.50) oder Vorkasse (zuzügl. DM 2.– Porto) auf Postscheckkonto Mchn., Kto.-Nr. 3403 13-807, Euroscheck oder Briefmarken.
REDYSOFT Software GmbH, Postf. 12 61, 8150 Holzkirchen [G]

MARKENQUALITÄT ZU UNGLAUBLICHEN PREISEN

VISA LSR600	Laserdrucker, 1,5 MB, 6 S./min	DM 4275.–
MICROPOLIS 1335	Winchester, 72 MB format., 28 ms, komplett mit SPEED-STOR	DM 1462.–
A.D.I. DM14+	14"-TTL-Monitor, bernstein, mit Fuß	DM 299.–
INTRA	14"-Flatscreen-Monitor TTL/RGB	DM 262.–
XT/AT-TASTATUR	102 Tasten, OMRON-ALPS-Switches, separater Cursorblock	DM 159.–
EIZO 8060S	Multisync-Monitor 14", 0,28 mm DOT	DM 1559.–
TVM Multiscan MD	Multisync-Monitor 14", Paperwhite	DM 594.–
D-Link	Netzwerkkarte	DM 404.–
ARCNET	Netzwerkkarte	DM 599.–
NOVELL	4-Platz-Software	DM 1502.–
93-Ω-Kabel	10 m mit BNC-Steckern	DM 39.–
Epromer 4fach	3 Brennspannungen, 2716-27512	DM 314.–
BABY-AT-BOARD	12,0 MHz, bis 4 MB max. on Board	DM 698.–

Versand per Nachnahme oder Vorauskasse auf Konto 14 134 603 Groß-Gerauer Volksbank (BLZ 508 925 00).
Alle Preise zuzüglich Versandkosten.
Komplettpreisliste gegen frankierten Rückumschlag.

M.L. WIEGAND EDV-VERSAND
POSTFACH 133 · D-6097 TREBUR 1
TEL. 0 61 47/77 37

Transmit Datasystems

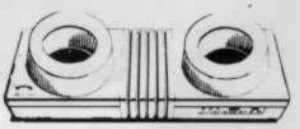
Ihr spezialisierter Partner für Drucker von

Wir lösen Druckprobleme an nahezu jedem Rechnersystem

Transmit
Datasystems GmbH
Fliederstraße 19–21
D-6454 Bruchköbel

Telefon: 0 61 81/7 70 84
Telefax: 0 61 81/7 98 20
Telex: 4 184 932 tru d

Datenfernübertragung



mit FTZ-Zulassung

Nur bei uns erhältlich:

DFÜ-Paket 21/23 für die gängigen Rechnersysteme bestehend aus :

— Akustikkoppler dataphon s21/23d (300, 600, 1200/75, 1200/1200 Baud halbduplex mit unserer KERMIT-Anpassung)
— KERMIT-Kommunikationsprogramm mit ausführlicher deutscher Anleitung
— Spezial-Anschlußkabel **nur 468,– DM**

Einzelpreise für Akustikkoppler
dataphon s21 d-2 nur **248 DM** — dataphon s21/23 nur **358 DM**

Bildschirmtext

NEU: BTX-Term
für PC **nur 268,— DM**

Paketpreis mit dataphon s21/23d **nur 578,— DM**

NEU: Jetzt auch Akustikkoppler und Modems mit 1200/2400 Baud voll-duplex mit FTZ-Zulassung.

Mailbox unter (02 41) 3 49 62 außerhalb der Geschäftszeiten

Außerdem erhalten Sie bei uns:
Rechner von SANYO + Drucker von Brother + Netzwerke von Molecular
Disketten von FUJI + Erweiterungskarten für PC + Logimouse C7
Software für alle Anwendungen
Händleranfragen erwünscht!

Geschäftszeiten: Mo—Do 9—13, 15—18.30 Uhr, Fr 9—13 Uhr

Andreas Krischer, Noppiusstraße 19
5100 Aachen, Telefon (02 41) 3 28 96

KRISCHER COMPUTERTECHNIK

Alexander Janson

dBase für Profis

Ausgefeilte Programmier-Tricks für Insider mit dBase II, III und IIIplus.
331 S., 89 Abb.,
geb. DM 68,–
ISBN 3-7723-8871-X

Dieses Buch ist randvoll mit wertvollen Hinweisen und Ratschlägen, die der Leser sofort in die Praxis umsetzen kann. Es ist klar und übersichtlich in Themenbereiche gegliedert und beginnt da, wo die übliche dBase-Literatur aufhört.

Das Franzis-Fachbuch
Ein Qualitätsversprechen
Franzis-Verlag GmbH, Karlstraße 37–41, 8000 München 2

mc 9-88-22

mc-MINIMARKT

BIETE AN SOFTWARE

Dongle oder Keykarte defekt? Wir helfen! Autocad 9.01 485 DM; Personal Designer V3.0 495 DM, Cad-Vance V2.02 465 DM, P-Cad V3.0 685 DM, Tango/Protel PCB Editor V3.12 285 DM, Win-Chart V1.03B 415 DM, Redboard Rel 5.0 465 DM, Cadkey V3.01 465 DM, PC2 V2.12 465 DM, Novell V2.1 390 DM, Eplan V3.1a 565 DM, Superpage V2.0 465 DM, MG-Cad V2.0 465 DM, ElCad86 V2.4 475 DM, Schema II V2.03 485 DM, Woody V3.1 435 DM, Producer 1/88 490 DM, PCDraft V5.3 435 DM, CADesi V4.4 515 DM, Windowmaker V2.6 425 DM, Scientex 365 DM. Info anfordern!
* Bitte Urheberrechte beachten *
! Deutsche Programmanleitungen !
*** Immer neueste Version ***
PC-Tools de Luxe 45 DM, CopyIIPC 25 DM, Norton Commander 30 DM, PC-Tools 35 DM, Flug-Simulator 40 DM, Norton Advanced Utilitie 40 DM, Copywrite 25 DM, VFeature de Luxe 45 DM, CopyIIPC Option Board 40 DM, Fastback 30 DM, Paintbrush 40 DM, DiskOptimizer 25 DM, News room 30 DM. Versand NN/V-Scheck +5 DM. Ing.-Büro Stock, Postf. 10, 7926 Böhmenkirch, ☎ 0 73 32/ 50 78, Btx 0 73 32 50 79 G

* **Stop * PD Software * Stop** *
MS-DOS XT/AT ab 3.– DM, z. B. dtsch. Prog., Girls, Rechng., Textverarbeitung, Datenverarbeitung, Grafik, Basic, Pascal usw. **Katalog gratis! 5. Auflage.** Achim Psenitza, Bocholder Str. 65, 4300 Essen 1, ☎ 02 01/68 22 89

Haftetiketten, 40 Größen, Info: Böhne, Försterweg 4, 3354 Dassel G

PUBLIC-DOMAIN-SOFTWARE für IBM-Katalogdisketten gegen 5.– DM. Funkcenter Mitte GmbH, Klosterstr. 130, 4000 Düsseldorf G

* **Aktuelles Softwareangebot** *
! Alles Original-US-Versionen !
Norton Advanced Utilitie V4.0 198 DM, CopyIIPC 4.01 98 DM, PC-Tools de Lux 138 DM, 1Dir Plus 178 DM, Norton Commander V1.02 125 DM, Copywrite 108 DM, Math-Cad 564 DM, Sideways 132 DM, PC-MOS 386 430 DM, Clipper S87 870 DM, Turbo-C V1.5 185 DM, Disk-Technician Plus 248 DM, Quattro 298 DM, Fastback 218 DM, Windows 386 338 DM, Copy Option Board de Luxe 298 DM, Disk Mechanic 168 DM, Makro Assembler V5.1 278 DM. Versand NN/V-Scheck + 5 DM. Ing.-Büro Stock, Postf. 10, 7926 Böhmenkirch, ☎ 0 73 32/ 50 78, Btx 0 73 32 50 79 G

Für QUICK-C und TURBO-C sgCURSES: Funktionssammlung zur Bildschirmsteuerung, kompatibel zu UNIX-Curses, DM 186.–. SG EDV-Beratung, ☎ 0 69/72 83 23

Ärger mit Wohnungsjahresabrechnung? Architekturberechn. o. belieb. Angebotsberechnungen? SOFTSIEGEL®BÜROTEXT 3.0 DM 396.– inkl. 200 S. Dok. löst das Problem.
Textbaustein zuladen + 2 Tasten drücken = automat. alles addieren! Über 20 Rechenbeispiele. dto. PLUS: 32 000 Adr./21 Feld./ Ser. Brief. DM 498.–. Demo: 4 Wohnungen 20.–, Hausprojektkalkul. 10 Geschosse 20.–. Verlag J. Leibrandt, Postfach 74 01 28, ☎ 0 69/65 65 68 G

* NC-PROGRAMMIERUNG *
Programmiersprache für MS-DOS-Rechner. Einfach und schnell für Werkstatt u. Ausbildung einsetzbar. Dialogführung und Grafikdarstellung. Durch PP-Schnittstelle an alle Steuerungen (Erodieren, Fräsen usw.) anpaßbar. Auch im SOURCE-CODE erhältlich.
☎ 0 71 83/74 14, ab 19 Uhr

Deutsche Anleitung f. PC-TOOLS deluxe DM 19.80.
☎ 0 52 41/5 67 47

MS-DOS-TOOLS, ein Muß für jeden Anwender, zu Preisen, die sich jeder leisten kann. Info kostenl. IBD Decker, Kalchreuther Str. 136, 8500 Nürnberg 10

* **Unglaublich, aber wahr!** *
* Über 3000 Disketten mit PC- *
* Freiprogrammen ab 5 DM *
* lieferbar. Der neue Katalog *
* ist schon da, kostenlos wie's *
* immer war. ☎ rund um die *
* Uhr 0 52 61/8 89 01, wagen *
* Sie's nur. Günther, Braker *
* Mitte 28c, 4920 Lemgo G

SUCHE HARDWARE

CBM-Matrixdrucker mit IEC-Bus bis 150.– gesucht. ☎ 0 89/ 85 55 78

Suche OPUS-Disc/Keyboard f. SPECTR. D. Pollmann, Firreler 39, 2954 Hesel

Suche in Deutschland günstige und bekannte japanische Hardware für Wiederverkauf in der Schweiz. Zuschr. unter mc 109 an den Verlag

Suche Akustikkoppler dataphone s21–23d o. ä. ☎ 0 21 59/8 01 48

BIETE AN HARDWARE

Verkaufe günstig: CP/M 2.2 Computer (MC-Computer + 68000 CO-Platine) m. viel Zubehör, Bücher u. Software. ☎ 09 51/7 11 55

Büronetzwerk 2 AT's Scanner, Laserdrucker, Softwarepaket, 8 Mon. (neu 70 000 DM) VB 22 000 DM. ☎ 0 69/44 30 00, 14–18 Uhr

mc-MINIMARKT

* **AT + 386 kompatible** *
* Baugruppen u. Fertiggeräte: *
* Power-ATO: **20 MHz Ows,** *
* **Speed 26.3.** Power-AT1: **16** *
* **MHz Ows, Speed 21.3.** Po- *
* wer-AT2: 13 MHz Ows, *
* Speed 17.3. Turbo-AT1: 10 *
* MHz Ows, Speed 13.0. Po- *
* wer-386: 20 MHz Ows, *
* **Speed 26.3.** *
* Ethernet Netzwerke: *
* 10 MBit/s. *
* Unterlagen anfordern: *
* G. M. E-Versand, *
* Postfach 10 01 01, *
* 4048 Grevenbroich, *
* ☎ 0 21 81/4 86 72 Ⓖ

Verkaufe **IBM Model 80** (8580; 70 MB; 20 MHz); IBM 8503=Monochrome Display; IBM 4869=5¼"-Floppy; IBM 4202 Proprinter XL; Software bes. geeignet für Baufa.; Komplettpreis VB DM 19 500.–. ☎ 09 51/2 63 94/2 74 98/6 20 72

Pinwriter P2200 dt. m. Zubehör 650.–; NEC Multisync I 900.–. ☎ 0 89/46 57 13, abends

---

NDR-Computer Systemauflösung! Z80 & 68008, 15 Karten, gr. Tast., Monitor bern., 19"-Geh., Teac-Lw. 2× 3,5", 1× 5,25", viel SW + Lit., auch einzeln. ☎ 0 71 62/39 98

HEWLETT-PACKARD Rechner 9830, mit 9867B Hard-Disk (14 Zoll) RS232, und Matrix ROM, DM 750. RENA100A Drucker, 132 Z./Zei., DM 300. ☎ 0 24 61/61 42 59

Siemens PC D/PC X, wahlweise für MS-DOS- oder SINIX-Betrieb DOS/SINIX-Orig.-Anwendersoftware; 1 HD, 1 FD, 1-MByte-RAM und Maus, wenig gebraucht, VB **3700.–** DM. ☎ 0 89/6 70 69 29

---

* **High Quality-Farbbänder:**
* für FX-80, 800 9.75 DM
* für FX-100, 1000 14.50 DM
* für LQ-500, 850 15.– DM
* für P6 14.80 DM
* für P7 16.– DM
* ab 10 St. pro Typ 10 % Rabatt
* **Disketten No-Name:**
* 100 St. 5,25", DS, DD 75.– DM
* 100 St. 3,5", 2DD 250.– DM
* Versand per NN + Porto
* Infoliste gegen Rückumschlag
* **EDV-Versand**
* R. Kreutzberg & M. Sprenger
* Gronaustr. 8, 5600 Wuppertal 2
* ☎ 02 02/8 07 66,
* ab 16.30 Uhr Ⓖ

● **Kamera** für Ossi und Monitor
● **Laborwagen**
Traumhafte Preise z. B.
● D. Multimeter ab DM 108,--
3 Stck. ab DM 98,--
● D. Multimeter TRUE RMS ab DM 450,--
● F. Generator ab DM 412,--
● P. Generator - Testbildgenerator -
● Elektron. Zähler ab DM 399,--
● Netzgeräte - jede Preislage -
● Meßkabel - Tastköpfe - R, L, C Dekaden
● Adapter - Stecker - Buchsen -
● Video-Audio-Kabel u. v. m.
Prospekt kostenlos
Händleranfragen erwünscht
HAMEG sofort ab Lager **HAMEG**
Bachmeier electronic · Danziger Str. 4 b
2804 Lilienthal ☎ 0 42 98 - 49 80
HAMEG sofort ab Lager **HAMEG**

XT-Komp mit 30 MByte Festplatte, 360 KByte LW, 640 KByte Ram, EGA-Karte + EGA-Farbmonitor + Div. Preis 2550.–. ☎ 07 21/57 94 41

DIN-A3-Plotter. Kein Spielzeug! **Bausatz** kompl. mit Gehäuse und Interface nur **DM 298!** Fertiggerät nur **DM 398!** Bauplan **DM 10!** Auflösung 0,1 mm, Geschwindigkeit ca. 70 mm/sec. Kostenloses Info bei Peter Haase, Weissenberger Weg 226, 4040 Neuss 1, ☎ 0 21 01/53 08 12, ab 17 Uhr Ⓖ

UNIX-Rechner, 1 MByte RAM, 12 MB Harddisk, 5½-Zoll-Floppy, Software, VB 2900.–. MC68000, 512 KByte, 2 Floppy, Sprache, Uhr, CP/M68K, Handbücher, Software VB 1111.–. M. Wörner, ☎ 0 89/9 30 23 82

IND-311/IEEE-4888 Controller/Bus Analyzer/16 Bit Digital TTL Input/Output/mit Soft 600 DM. ENHANCED VCA321 250 DM. CGA/Mono VCA 351 120 DM. Modem COM 322/V22 200 DM. Alle NEU/Addonics/SERTEK. ☎ 0 89/30 69 31 52/6 70 54 45

MC 68010, 2 LW, Z80-EPROM-Uhren-Terminator-Karte, org. Software: 0S9/68K u. CPM/68K. VB 2000 DM (eventuell auch einzeln). ☎ 02 14/2 73 42 (werktags erst ab 18 Uhr)

Apple II+ Orig., 2 LW, 80-Z, Pal-Farbkarte, 6522-Parallel, ca. 50 Disketten + div. Zubehör, VB 800 DM. ☎ 0 21 95/33 95

Achtung! Verk. SIEMENS PCD-2T! 7 Mon. alt m. Gar., inkl. Farbmonitor, 40-MB-Festpl., MS-DOS 3.2 MS-Windows, GW-Basic u. Literatur; Preis VS. ☎ 0 81 46/6 07, ab 17 Uhr

---

SAM 68k (640 KByte), 2 Teac FD 556, frei progr. ACS Tastatur, Netzteil, neuwertig wegen Systemwechsel günstig zu verkaufen. ☎ 0 23 92/6 94 94

Z80-Einplatinen-Computer inkl. STI 150 DM, CGA-Karte 50 DM, 8 RAMs (256 KByte × 1, 150 ns) 100 DM. ☎ 0 21 73/5 05 06

---

* **SEL-TELEX** gebraucht: **1200.–** m. FTZ zul. & Lochstr.-L&S. * **Schneider CPC 464** mit 3"-Z.-Floppy und 30 St. Disketten, Farbmonitor mit Netzteil u. Joystick & CPM!!! nur **695.–** * **AMIGA-500,** 1 MB, 2 Laufwerke + Monochrom-Monitor **1999.–** * **12-Zoll-BAS-Monitor** NEU für C64, Apple, IBM, CGA, nur **160.–** * **12-Zoll-Philips-Monitor** wie oben auch NEU, nur **170.–** * **Video-Camera** SW kompl. **440.–** * **Video-Player** NEU, nur **490.–** * **180 Z./Sec. Drucker** DIN A3 mit par. Schnitt. geb. **290.–** * **Drucker-Puffer** 256-KB **384.–.** Versand: per NN zuzgl. Porto, **KOLTER-ELECTRONIC, Postfach 11 27, 5042 Erftstadt, Fax.: 0 22 35/7 20 48** Ⓖ

Softwaregesteuerte Schnittstellen-Umschalter

PortSaver
erweitert jeden Computer auf 8 oder mehr E/A-Kanäle
1 Rechner an 8 Geräten: Modem, Drucker, Plotter, Terminal, Datenleitung...

PortCollector
8 Rechner an 2 Geräten: Drucker, Plotter, Datenleitung... mit integriertem Spooler

Umschaltung/Steuerung durch einfache ASCII-Steuersequenzen
Automatische Erkennung der Übertragungsgeschwindigkeit
Keine Bedienung notwendig, einfach anschließen — fertig

Herstellung und Vertrieb:

ROHNER-EDV-Elektronik
Schelmenhofstraße 10, D-8950 Kaufbeuren
Tel. (0 83 41) 8 10 54, Tx. 5 41 290 roedv

Preise wie im Paradies!

EPSON

LQ-500	798,-
FX-850	1048,-
FX-1050	1298,-
LQ-850	1348,-
LQ-1050	1748,-

Star

LC-10	548,-
LC-10 Color	648,-
LC-24-10	798,-

Toshiba

T1000	1998,-
T3200	9398,-

Sharp

PC-4521	4398,-

NEC

P2200 758,-

CSF P2200	198,-
Multisync GS	498,-
Multisync II	1298,-
Multi.Plus	2098,-
Multisy. XL	4398,-

Plantron

PT-AT Tower	2448,-
Der Baby AT Tower: PT-286	3778,-

CCP

Baby AT	2248,-
Baby AT/64	3398,-
XT mit 10MHz	1098,-

Commodore

Amiga 2000	
inkl.Mon.1084	2598,-
PC-10 III	1665,-
PC-10 III 1/30	2148,-
PC-10 III 2/20	2198,-
PC-10 III 230	2298,-
PC-20 III	2398,-

Olivetti

M240/55G	3178,-
Aufpreis 20 MB	598,-
M290(20MB)	5798,-

Software

dBase III + dt.	1348,-
Word 4.0	938,-
Turbo Pascal	238,-

Seagate

20 MB Kit	548,-
30 MB Kit	588,-
40 MB (40ms)	718,-
40 MB (28ms)	798,-
ST-125-0(40ms)	498,-
ST-157R-O	828,-
80 MB (28ms)	1198,-

EGA/VGA

Mitsubishi	1298,-
Hitachi M 560	1248,-
EGA Wonder	398,-

Genoa

EGA HiRes+	378,-
Super VGA	578,-
VGA HiRes	798,-

Computer Discount 2000 GmbH MC/9/88

Hinter der Bahn-5403Urmitz-Bahnhof-Tel.02630/6031-TTX(17)2630915-Fax 02630/84366

---

mc

Anzeigen-schluß-termine

Heft 11/88 vom 24. 10. 88
Anzeigenschluß 19. 9. 88

Heft 12/88 vom 28. 11. 88
Anzeigenschluß 24. 10. 88

Heft 1/89 vom 27. 12. 88
Anzeigenschluß 21. 11. 88

Inserieren lohnt sich immer

PROLOADER II professional

★ bearbeitet vollautomatisch bis zu 120 Disketten
★ ausgereiftes deutsches Betriebssystem (AGATHE II)
★ Anschluß an jeden XT/AT
★ 1 Jahr Garantie

** **Komplettpreis DM 4950.-** **

EXPRESS SERVICE
Astrid Günther
Braker Mitte 28 ★ 4920 Lemgo
Tel. 05261-88901 rund um die Uhr!
– Händleranfragen erwünscht –

mc-MINIMARKT

BIETE AN HARDWARE

NDR-Comp. CPU Z80 = 30.–, GDP64K = 100.–, Key = 30.–, IOE = 30.–, Hex/O = 45.–, Ehex 2 = 20.–, POW5 = 15.–, Trafo 2 = 25.–, Cas = 20.–, IOE 2 = 50.–, ESPS 2 = 80.–, SPS-Kurs = 100.–.
☎ 0 61 46/67 49, ab 16 Uhr

IBM PC XT (SFD), 640 KByte, 20 MB FP, 1 LW, div. IBM-Adapter, MF-Tast. IBM EGA m. 256K + Grundig BGC 36, 1½ J. 4900 DM.
☎ 0 21 59/8 01 48

Reelle Sonderangebote von dem Modular und Laptop-Spezialist: mc-**modular** 10/12 MHz Bausätze, Fertiggeräte schon ab 1850.– DM * **Laptops's** in div. Ausf. 1600.– DM * **Portables** at mit 20–80 MB – 48 m Festplatine, 12–24 MHz ab 2200.– DM * quadran **multi-vga** ega-set inkl. Monitor, Karte 1400.– DM * **Laserdrucker** 10B 100 % hp plus kompatibel, 1a-Qualität, 3900.– DM. **Beratung** wird bei uns **groß** geschrieben: ☎ **0 62 28/7 60** ⌕ **apcom** Ⓖ

Netzstabilisator WS-6, 1 KW, W. & G., zur Vermeidung v. Datenverlusten, 300 DM. ☎ 0 24 61/61 42 59

RESTPOSTEN Eproms, Drams, Intel 2764, 27128, 27256, AM2764, F2764.
☎ 0 74 86/73 79 Ⓖ

NDR-Computer CPU68K + Z80 auf einem Bus CP/M68K + CP/M2.2, Prommer HCopy Centronics, RAM 480 KByte, Turbo-Pascal C-Compiler Basic usw. + Monitor kompl. 800 DM ab 18 Uhr.
☎ 0 52 08/14 35

* AT 286 Komplett-Paket *
* DM 3198.– *
* – 6/12 MHz, 0 Wait-State, *
* 512 KByte bestückt/4 MB on *
* Board, Uhr/Kalender, akku- *
* gepuffert *
* Herkules-Karte *
* Ser.-/Par.-Schnittstelle *
* 101 Key-DIN-Tastatur *
* 1,2 MB Laufwerk *
* 20 MB Festplatte *
* FDD/HDD-AT-Controller Ge- *
* häuse mit Reset/Schl.Sch. *
* Deutsches Handbuch *
* 1 Jahr Garantie *
* R L - Computer GmbH *
* Hunsrückstr. 25 *
* 4020 Mettmann *
* ☎ 0 21 04/8 24 26 o. 5 29 31 *
* andere Konfigurationen, so- *
* wie Karten, Tastaturen, usw. *
* auf Anfrage *
* Preisliste anfordern Ⓖ

RAM CHIPS: TI 4256-12, CMOS HM 51256-100 zum Spitzenpreis. Hitachi LM 215B LCD 480 × 128 + Controller 500 DM. ☎ 0 52 61/57 65

Turbo XT 12 MHz, V20, HD 32 MB, AT Keyb., Maus, Multi I/O, Akku, Uhr, 60 Disketten, Softw., Disk-Box 1850.–. ☎ 0 95 21/89 63, ab 17 Uhr

MC68000, 1024 KByte, 2 LW 3,5", Netzteil, Geh., Tastatur, Mon. CPM68K, XED, Fortran, C, Z80-Emul. usw. VHB 2200 DM.
☎ 0 71 23/7 13 41, ab 18 Uhr

NDR-Klein-Computer. Verkaufe **ACRTC** preiswert! ☎ 0 25 42/61 14

mc-CP-System inkl. Sw 200.–. **E-Lab** CPU1.3 + FDC2.2 + 8"-Lw+Sw 600.–. **18 SIG/M-Disk** (5" Kaypro 4) 90.–. ☎ 0 75 33/18 99

HP Laserjet Postscriptkarte Sammelbestellung aus USA ca. DM 3000. ☎ 02 51/86 42 61, ab 20 Uhr

Verkaufe NDR 68K + Z80 im GEHÄUSE BUS4 SBC3 Z80 CPU68K KEY SOUND SPRACHE SER CENT CAS GDP HCOPY PROMER RAM256 DR128 5×ROA64 BANK FLO3 MAUS TAST MONGRÜN SOFT Z80 Grund Basic CPM2,2 WS DP MULTI M80 280 MBASIC BIOSPAT SOFT 68K Grund R2 Basic Jogi Ham Jados u. Systemados Sprache Sound Rubapaint MCopy Pet Text ey Buchhaltung Hexey Dosey VB 2500.– DM + UNTERLAGEN.
☎ 0 81 31/8 22 98 ab 17 Uhr

Handy-Scanner mit Texterkenn. 550 DM, ☎ 0 69/38 81 44

* **Commodore 128** + Joystick, Monitor + Kabel, NEU **659.–**
* **Laufw.** für PC 5,25/360k **197.–**
* **Controller** für PC/360k **39.–**
* **Herkules-Graph.** & Par. **89.–**
* **2D-Disketten** MARKENWARE ab 200 NEU verp. stk.: **–.79**
* **Plotter** DIN A6, 4 Farb. mit Stiften, Ser./Par. nur: **490.–**
* **TERMINAL** 50–9600 Bd. mit Bildschirm und V.24 ab **590.–**
* **HP-8916** 256K-Karte geb. **280.–**
* **GE-Drucker** par. Thermo und Normalpapier kompl. **290.–**
* **MAUS** für PC NEU nur: **88.–**
* **Drucker-Puffer** 1 MB **1280.–**
Versand: per NN zuzgl. Porto. **KOLTER-ELECTRONIC, Postfach 11 27, 5042 Erftstadt, Fax. 0 22 35/7 20 48** Ⓖ

Gewerbl. Anzeigen sind mit Ⓖ gekennzeichnet.

MZ-80B (SHARP) mit FDOS/CPM 2.2, 2 Graphik-Ebenen, allen Interfacen (LPT/Centr./Floppy/IEEE-488/2× ser./8255-PIO), Doppel-Floppy (2× 360 KByte) und viel prof. Software (dBase, TurboPascal, BASIC-Comp., etc. + 1 Meter Handbücher) gegen Höchstgebot. Dr. Hardtke, ☎ 0 42 36/13 33 n. 20 h

EPROMSIMULATOR für 2716 bis 27512. Fertigplatine: **DM 278.–,** Komplettgerät: **DM 428.–. Info bei: HüWiCo,** Am Siepen 17, 4630 Bochum, ☎ 02 34/36 12 06 Ⓖ

* SCHALTNETZTEIL im Gehäuse * 5 V/12 A, 12 V/4 A, –12 V/300 mA: 79 DM. Gerloff Elektronik, ☎ 0 51 46/86 81 Ⓖ

TRANSPUTER-Karte TEK4/8 1MBCMOS-RAM. ☎ 0 69/4 07 91 89 o. 0 61 02/3 78 83

mc68010 mit SASI-Interface, Dokumentation, funktionsgeprüft durch DSM, Software, VB 1300.–.
☎ 04 21/87 69 92

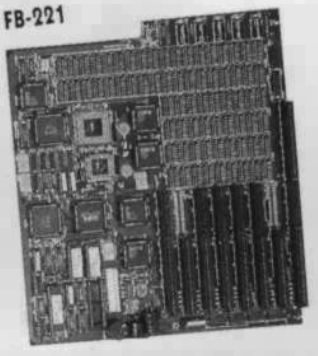
TELEX-ADAPTOR mit ZZF-Nummer für Existenzgründer und alte Hasen mit V24/V28 vernetzbar mit einer oder mehr elektr. Schreibmaschinen oder PCs =. **ELEKTRONISCHER FERNSCHREIBER** (mtl. DM 96 Postgebühren) zum SUPERPREIS von **DM 3500.–** inkl. MwSt. TECHNOCHEM HGMBH, Frankfurt,
☎ 0 69/74 98 98, Tlx. 4 170 532 Ⓖ

Magnetkartenleser.
☎ 05 71/4 98 56 Ⓖ

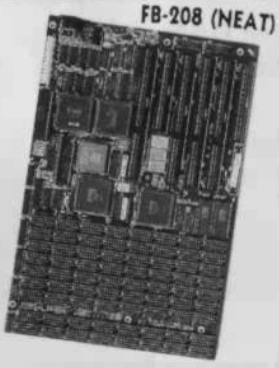
Intel 80386 25 MHz, System 302. Sammelbestellung aus USA ca. DM 4500. ☎ 02 51/86 42 61, ab 20 Uhr

We provide best price/performance mainboard & add-on cards.

FB-221



FB-208 (NEAT)



SPECIFICATIONS: FB-208 286 MAIN BOARD (NEAT)
* 10/12/16/20MHZ HARDWARE/SOFTWARE SWITCHABLE
* SUPPORTS 16MHZ 0 WAIT STATES WITH 100NS DRAM AND 12MHZ 0 WAIT STATE WITH 150NS DRAM.
* PAGE/INTERLEAVE MEMORY CONTROL.
* 2MB/8MB DRAM ON BOARD.
* OPTIMIZED FOR OS/2 OPERATION

WE ALSO SUPPLY:
* ADD-ON CARDS FOR PC/XT, AT
  – MULTI I/O CARD
  – SERIAL/PARALLEL CARD
  – RS-232
  – DRIVE CARD
* 8088, 80286, 80386 FULL SYSTEMS ARE AVAILABLE.

IBM PC/XT/AT are registered trademarks of International Business Machines Corp.

FB-221 386 MAIN BOARD
* HARDWARE/SOFTWARE SPEED CONTROL
* 16/20MHZ 0 WAIT STATE OR 20/24MHZ SWITCHABLE 0 WAIT STATE.
* 80387/80287 OPTIONAL
* 2MB/8MB DRAM ON BOARD.
* OS/2, WINDOW/386 UNIX, PCMOS 386

• CONTACT US TODAY FOR FULL DETAILS!

MANUFACTURER & EXPORTER
FANN BAO ELECTRONIC CO., LTD.
P.O. BOX 78-40, TAIPEI,
10TH FL., NO. 206, FU-YUAN ST.,
TAIPEI 10598 TAIWAN, R.O.C.
TEL: 886-2-7668191-2 TLX: 16288 FANNBAO
FAX: 886-2-7566394

Netzteil defekt? Wir reparieren schnell!
Reparaturservice für alle Fabrikate

ATS
All Type Electric Service

Sandweg 18
7530 Pforzheim
(07231) 44638

mc-MINIMARKT

286 Speed-Karte nur 798.– DM, 10 MHz und 8 KByte Cache-Speicher, **Ihr XT+Comp. schneller als AT!** Hdl.-Anfragen erwünscht! **FSM-TROGISCH,** Gartenstr. 10, 3051 Suthfeld, ☎ 0 57 23/8 19 46 Ⓖ

SCHRITTMOTORSTEUERUNG XYZ-Achsensteuerg. f. Computer mit Parallelport. Mit Netzteil und 3 Schrittmotoren **DM 269.– BOHRPROGR./C64 DM 98.– Schrittmotor DM 29.–.** Info DM 2. **PME,** Hommerich 20, 5216 Rheidt, ☎ 0 22 08/28 18 Ⓖ

HARDWARE-MESSWERT-ERFASSUNG für fast alle Computer (siehe mc-Quickie). **L. Bockstaller** Ⓖ

*** Steckfertig und preiswert: Transputer Board TEK 4/8 nach CT 1087 für PC, AT, ATARI ST. Ausstattung, Preis. ☎ 0 52 61/57 65

Festplatten-Subsystem mit Adapter dazu (ohne Festplatte). Dr. Hardtke, ☎ 0 42 36/13 33, n. 20 h

Apple II: TZ-Quattro CAD-Programm und neue Hardware: PC-Transporter, (IIGS) RAM-Karten Z80 H, spez. //c und GS-Karten u. a. von AE, Cirtech u.v.m.! Info unter ☎ 0 30/7 45 00 06

KONTAKTE

Wir suchen Menschen, die Programmierer werden wollen. Durch Fernkurs zu fundierten Kenntnissen als BASIC-, PASCAL- oder COBOL-Programmierer. Ohne besondere Vorbildung werden Sie Computer bedienen, Programmsprachen beherrschen, Programme entwickeln, testen und anpassen können. Als zukunftsorientierte berufliche Weiterbildung oder interessante Freizeitbeschäftigung. 52 weitere Fernkurse. Kostenlosen Studienführer und Probelektion anfordern. Kein Vertreterbesuch. **Studiengemeinschaft Darmstadt, Abt. 36/02, Postfach 41 41, 6100 Darmstadt,** ☎ **0 61 57/80 66** Ⓖ

68000, 6809, Dipl.-Ing. entwickelt Hard- und Software, EPAC-Systeme, intel. Interfaces, IEC-Bus. ☎ 0 69/39 38 84

HX-20-CLUB und PC-PROFI-GRUPPE c/o Jürgen und Susanne Miersch, Hindenburgd. 58a, 1000 Berlin 45

Gewerbl. Anzeigen sind mit Ⓖ gekennzeichnet.

TAUSCH

Festplatte Seagate, 31 MB, neu. Tausch gegen Autocad 9.01. ☎ 0 61 34/6 57 71

VERSCHIEDENES

Raum Aachen/Köln: Diskettenkonvertierung CP/M nach **MSDOS** und umgekehrt für 25.– DM per Disk. **Siegfried Müller, Hauptstraße 98, 5163 Langerwehe** Ⓖ

MAC-TOS-DOS-OS2-9: Konvertieren v. Disketten 3,5"–5¼"-HD. ☎ 02 51/86 42 61, ab 20 Uhr

* BESTÜCKUNG * MONTAGE * PRÜFUNG * Layout- und Leiterplattenservice * Vom Muster bis zur Serienproduktion. Jede Leiterplatte maschinengelötet, sichtkontrolliert und blitzblank ultraschallgereinigt. Ein schnelles Preisangebot erhalten Sie nach Einsendung von Bestückungsplan und Stückliste oder Bestückungsmuster. gk-electronics, Hofrichterstr. 34, 5000 Köln-80, ☎ 02 21/63 98 97, Fax: 02 21/63 42 37. Produktion: Adamstr. 37, ☎ 62 50 85 Ⓖ

Das Postspiel: GALAKT.-STAR-WARS. Bestehen Sie den Überlebenskampf im Weltall. Info von Astro-Data-Club, Postfach 943, 6400 Fulda

Verdienst m. eigenem Computer 24seit. inter. Info, 21.30 DM (davon 1.30 Porto). G. Hein, In der Worth 12, 3155 Edemissen

* Sonderposten-Copy-Cards * behebt BACKUP-PROBLEME geschützter Software: Info, ☎ 08 21/51 99 98 Ⓖ

mc Jg. 1981–1985 (52 Hefte), neuw. Zust., kompl. DM 120 VHB. ☎ 0 75 31/7 69 40 (nach 18 Uhr)

** PAL/GAL Programmierservice ** Info anfordern von Andreas Haack, Heilmannring 59a, 1000 Berlin 13. Kein Ladenverkauf Ⓖ

Service Manual Amiga 2000 für DM 100.–. ☎ 0 89/91 76 59, abends

EDV-Literatur
Liste kostenlos: H. Weidinger, Postf. 21 05 46, 8500 Nürnberg 21 Ⓖ

Eine eigene Firma
muß kein Wunschtraum bleiben. Neuartige Wirtschaftszeitschrift liefert X Beispiele, wie Sie erfolgreich Ihr eigener Chef werden. Gratisinfo von: Die Geschäftsidee, Th.-Heuss-Str. 4/MC835, 5300 Bonn 2

Alexander Janson

MS-DOS kompakt

Versionen 2.0 bis 3.3
184 S., geb., DM 38,–
ISBN 3-7723-7602-9

Dieses Buch ist ein Nachschlagewerk, das alle praxisrelevanten Informationen in der komprimierten Form eines Kurzhandbuches darbietet, so daß Sie es als Arbeitshilfe unmittelbar am Computer benutzen können. Zu jedem Kommando finden Sie wichtige Hinweise, erläuternde Beispiele und Verweise auf ähnliche Kommandos.

Das Franzis-Fachbuch
Ein Qualitätsversprechen
Franzis-Verlag GmbH, Karlstraße 37–41, 8000 München 2

mc 9-88-35

Eprommer//Eprommer
mit ext. Textoolsockel im Gehäuse., Editor, Programmierspannung: Vpp 25V; 21V und 12.5V Programmiert 2716 bis 27512 auch C Typen Lieferumfang: Steckkarte ext. Epromsockel, umfangreiche Software Beschreibung und ein Programm zum einlesen von Intel HEX-Files.
nur **248 DM**

Novell® Netzwerke
Novell Advanced Netzwerksoftware für 286/386 Fileserver v2.11 (SFT und ELS Software auf Anfrage) Auch weiteres Zubehör wie *BNC Stecker *BNC T-Verbinder *Koax-Kabe RG 58 CU Preise auf Anfrage. Netzwerkpreisliste anfordern

Netzwerkkarte Arcnet® kompatibel 2,5MB/S **598 DM**
Aktive Weiche 598DM Passive Weiche 98DM
TEAC 55GFR 1,2MB **275DM Thin Ethernet** Karte10mbs **598DM**
WD Ethernet Karte 10mB/s IEEE 802.3 nur 648,DM
TEAC 1,44 MB und 720 KB Laufwerk 275 DM Hanbuchje 15DM

UEDING electronics
Holtewiese 2 Tel. 02373 63159 Händleranfragen
5750 Menden 1 DFÜ 02373 66877 8n1 erwüscht
inh. Gregor Ueding BTX 02 3736 3159 9/88

Macintosh

Apple II

MS-DOS

HARDWARE SOFTWARE ZUBEHÖR BÜCHER
Spezialkatalog für ☐ Apple II , ☐ Macintosh , ☐ MS-DOS
Fordern Sie unter Angabe Ihres Rechnertyps den entsprechenden **Gratiskatalog** an !

pandasoft Dr.-Ing. Eden
Uhlandstr.195 D-1000 Berlin 12
Tel.: 030 / 31 04 24
Telex: 185 859

BITTE SCHICKEN SIE MIR IHREN GRATISKATALOG ZU!
Name:
Adr.:
Rech.Typ:
MC

mc-MINIMARKT

VERSCHIEDENES

Leiterplatten: Sofortdienst für Muster, Layouts, Entwicklungen. ☎ 0 77 02/22 88 u. 18 84 G

Durchkontaktierte Leiterplatten; Oberfläche Pb/Sn ungeschmolzen, Klein- und Mittelserien, Muster in wenigen Tagen, auch einseitig. **Bestückungen – Layouts.** M. Ernst, Elektron. Systeme, Postfach 8 41, 5800 Hagen 1, ☎ 0 23 31/40 30 07 G

STAUBSCHUTZHAUBEN – EIN PROBLEM WENIGER – z. B. für IBM-PC und Kompatible, zweiteilig (Zentraleinheit mit Monitor und Tastatur). NUR DM 40.– zzgl. Versandk. Gesamtpreisl. DM 3.– Briefm. **K * S-VERSAND** K. Schellhammer, Dorfstr. 26, 2061 Kleinwesenberg, ☎ 0 45 33/35 66 Fax 0 45 33/52 87 G

** Bohren von Leiterplatten ** – Präzise und termingetreu – Klein-, Mittel- und Großserien Gerloff-Elektronik, ☎ 0 51 46/86 81

mc, kompl. Jahrgänge 81–88, 150.– DM. ☎ 0 25 01/28 78

Wir haben ein 68008-System mit 4 MByte Adreßraum entwickelt. Es ist ein Slot-Computer, der universell ausbaubar ist. Praktisch jedes Betriebssystem kann angepaßt werden. Ein eigenes in Modula 2 geschriebenes Betriebssystem wurde entwickelt. Außerdem gibt es einen in Assembler geschriebenen Modula 2 Einpass Compiler. Wir suchen Firmen und Leute, die sich mit Geld und Know-how an unserer Firma beteiligen. Außerdem vertreiben wir bereits ein 68000/68010/68020 Europakartensystem für gehobene Ansprüche, das in der Leistung besser als die IBM-PS/2-Serie ist und außerdem billiger. LB Computer, Postfach 12 53, 8425 Neustadt/Donau

Orig.-**Cannon**-Hauben, 37- und 50pol., à **DM –.75.** ☎ 0 72 22/8 16 35

FLOPPY AM 6502-EMUF und anderen Rechnern. Bauanleitung mit Schaltplan u. 6502-Routinen geg. DM 20.– VK od. NN. 3,5"-Diskette + DM 10.–. M. Meyer, Adamstr. 48, 1000 Berlin 20

Eigener Computer bringt Geld, 24seitige interessante Info, 20 DM + 1.40 Porto. G. Hein, In der Worth 12, 3155 Edemissen

MS-DOS für Fortgeschrittene

Das große MS-DOS-Profi-Arbeitsbuch

MS-DOS intern – für den Profi in allen Versionen aufbereitet. Mit PC-DOS 3.3. Von D. Smode. 2. Aufl., 448 S., 71 Abb., Tab. und Listings, geb. DM 68,– ISBN 3-7723-8682-2

Im Buch wurden alle Versionen bis einschließlich PC-DOS 3.3 berücksichtigt. Das Haupt-Augenmerk liegt nicht auf der Mensch/MS-DOS-Schnittstelle, sondern im Bereich Anwenderprogramm/MS-DOS.

Das Franzis-Fachbuch

Ein Qualitätsversprechen

Franzis-Verlag GmbH, Karlstraße 37–41, 8000 München 2

mc 9-88-20

Auftrag für Gelegenheitsanzeige in mc

Anschrift: Franzis-Verlag GmbH, Anzeigenabteilung, Karlstraße 41, 8000 München 2

Name und Adresse _____

_____ _____

Unterschrift/Ich zahle sofort nach Rechnungserhalt. Datum

☐ Preis für private Gelegenheitsanzeige je Druckzeile 4.– DM. Chiffregebühr je Anzeige 8.– DM.

☐ Preis für gewerbliche Gelegenheitsanzeige je Druckzeile 14.82 DM (dürfen nicht unter Chiffre erscheinen). Bitte beachten Sie, daß **Preise in Anzeigen die MwSt.** enthalten müssen.

mc-programmbörse
☐ suche Software
☐ biete an Software

mc-minimarkt
☐ suche Hardware
☐ biete an Hardware

☐ Tausch ☐ Verschiedenes
☐ Kontakte ☐ Chiffre-Nummer
☐ vollständige Adresse
☐ Telefon

Ich wünsche folgenden Text in der nächstmöglichen Ausgabe zu veröffentlichen:

(Zutreffendes bitte ankreuzen)

Bitte mit Schreibmaschine oder in Druckschrift in Groß- und Kleinbuchstaben ausfüllen.

INSERENTENVERZEICHNIS

Abor 31, 32, 33
Adcomp 29
AD Computer 11
Ahlers 145
ASM 145
A.S.S.-Ware 145
ATS 178

B & P 148
Blancke 144
Bockstaller 145
bsb Datentechnik 144

CCE 145
CMTE-FAST 149
Computer Discount .. 177
Computer 2000 ... 16, 17
Conex 31, 32, 33

Datacomm 174
Datapro 146
Datatronics 158
Ditfurth v., Leasing ... 147
Dobbertin 149
Dodytech 159
DSM 12, 13

ECO-Institut 30
Ecomedia 26
ees GmbH 147
Elco 24
Elcoprint 156
Elektronikladen 35
Elwo 145
Engelhardt 148
Erb-Verlag 163
Express Service 177

Fann Bao 178
Franzis 141, 167, 176, 179
Fujitsu 7

GAI 147
Gorny 148
Graf 21

Haase & Menrad 144
Habersetzer 156
Hanser-Verlag 152
Holco 160
HS-Computer 149

IBP 171
Indutronic 3, 169
Intec 147
Intra 101
iSystem 172

Kolter 148
Krischer 176
Kwem 23
Kyocera 39

Lauterbach 170
Laytronic 148
Lead Year 161

Mathes 164
Mayon 148
MCI 92, 93, 183
Memory Electronics .. 144
Meyer E.W. ... 31, 32, 33
Micom 144
MicroTech 171
Monterey 162
MoVe 149
MPK 147
Müller, Dr. Gert 146
Multisoft 148

Nanlien 168
NEC 41

Pandasoft 179
Physik Instrumente .. 149
Plantron 2
PublicSoft 27

QT-Software Kuthe .. 145

Radio RIM 168
Ranfft 148
Rheintec 173
Rhothron 149
Röntgen 144
Rohner 177
Rose 146
RWL 153

Schlaps & Partner ... 147
SE Spezial Elektroni .. 9
Shamrock 139
Sintronic 25
Stage 15
Stone 157
Strässle 155

Tandon 36, 37
TecSys 147
Tetzlaff 147
Toshiba 61
Transmit 175
TSS-Schmitz 148

Ueding 179
Unisoft 145

Vobis 184

Walter 173
Weber 144
Wegner 145
Wiegand 175
Wiesemann & Theis .. 170
Wolff 146

mc

Die Mikrocomputer-Zeitschrift

REDAKTION

Anschrift:
Franzis-Verlag GmbH
Karlstraße 37–41, 8000 München 2
Postfach 37 01 20, 8000 München 37
Sekretariat: Rita Schieser
Telefon: 0 89/51 17-3 54
Telex: 5 22 301
Telefax: 0 89/51 17-3 79

Chefredakteur:
Dipl.-Math. Ulrich Rohde, verantw.
(Anschrift der Redaktion)

Stellv. Chefredakteur:
Dipl.-Ing. (FH) Ulrich Kruppe

Redaktion: Horst Brand, Dipl.-Ing. (FH) Wolfgang Hascher (fl), Aktuelles: Reiner Schönrock, PC-Technik: Dipl.-Ing. (FH) Dieter Strauß

Ständige Mitarbeiter (zu erreichen unter der Anschrift der Redaktion): Helga M. Schmidt, Günther Sternberg

Korrespondent:
USA: Ray Duncan

Art Direction: Dipl.-Des. (FH) V. Hilbel

Layout, Grafik, Herstellung:
Josef Wurzinger

Labor: Neucom Electronic GmbH

Software-Service:
Shamrock Software-Vertrieb,
Telefon 0 89/59 54 68

Sonderdrucke: Jakob Wintersberger

Lizenzen: Georg Geschke, Königstr. 8, 3000 Hannover 1, Tel. 05 11/34 53 62

Gesamtherstellung: Franzis-Druck GmbH, Karlstraße 35, 8000 München 2, Telefon 0 89/51 17-1

VERLAG

Anschrift:
Franzis-Verlag GmbH
Karlstraße 37–41, 8000 München 2
Postfach 37 01 20, 8000 München 37
Telefon: 0 89/51 17-1
Telex: 5 22 301
Telefax: 0 89/51 17-3 79
Postgirokonto München 5758-807

Geschäftsführer:
Peter G. E. Mayer,
Michael-Alexander Mayer,
Dr. Harald Wiebking

Anzeigenleitung: Dietger Kötter, verantw.
(Anschrift wie Verlag)

Anzeigenverkaufsleitung: Hans-Joachim Hecht (-3 86)

Disposition: Ingrid Daschner (-2 97)

Vertriebsleitung: Herbert Barnehl

Abonnement: Christa Fischer (-2 40/-2 79)

Handelsverkauf: Dietlind Stauder (-2 04/-2 83)

Bezugspreise Inland: Einzelheft 7,– DM, Jahresabonnement 70,– DM; Vierteljahresabonnement 18,50 DM. Studenten, Auszubildende und Rentner erhalten das Jahresabonnement gegen Nachweis günstiger. Der Versand ist im Abonnementspreis eingeschlossen. In den Preisen ist die gesetzliche Mehrwertsteuer in Höhe von 7% enthalten.

Die mc erscheint monatlich, jeweils montags am Monatsanfang bzw. am Ende des Vormonats; im 8. Jahrgang.

ISSN 0720-4442
Vertriebskennzeichen B 7745 E

VERLAGSVERTRETUNGEN

Anzeigenvertretung Inland:
Baden-Württemberg: Ulrich G. Felger, Honoldweg 27, 7000 Stuttgart 1, Tel. 07 11/63 27 18

Bayern: Elfie Rusch, Münchner Verlagsvertretung, Sperberstraße 23, 8000 München 82, Tel. 0 89/4 30 73 32

Berlin: Rainer W. Stengel, Bischofsgrüner Weg 91, 1000 Berlin 46, Tel. 0 30/7 74 45 16

Hessen: Günter Junne, Victor-Achard-Str. 30, 6380 Bad Homburg v. d. H., Tel. 0 61 72/3 33 94

Norddeutschland: Lita Lange, Impulse medien service GmbH, Holtenklinker Str. 9, 2050 Hamburg 80, Tel. 0 40/7 24 20 37/38

Nordrhein-Westfalen: Bandelow + Partner GmbH & Co. KG, Herr Werner Bandelow, Dellestraße 36, 4000 Düsseldorf 12, Tel. 02 11/20 14 64

Anzeigenvertretung Ausland:
Schweiz: Exportwerbung AG, Kirchgasse 50, CH-8024 Zürich, phone: 01-47 46 90, telex 8 12 765

Großbritannien: Martin Geerke, 4, Friary Hall (Flat 3), Friary Road, South Ascot, Berks SL5 9HD, U. K., phone: 9 90-2 86 49, telex: 858 328 EUROAD

Japan: ABC Enterprises Inc., Heinz W. Kuhlmann, 7-4, Ohyama-cho, Shibuya-ku, Tokyo 151 Japan, Tel. 4 85-29 61-3, Fax 4 66-07 09

USA: DemoNet Inc., 7330 Adams St. Paramount, CA 90723, phone: (213) 408-0999, telex No. (910) 321-3026

France: Agence Gustav Elm, 41, Avenue Montaigne, 75008 Paris, phone: 01-47 23 32 67

Italia: Rancati advertising, Milano San Felice Torre 5, I-20090 Segrate, phone: 02-7 53 14 45, telex: 3 11 250 PPMII

Belgien: ECI/United Media Int. S.A., Avenue de la folle chanson, 2 bte 7, 1050 Bruxelles, Tel. 02/6 47 31 90, Telex: 6 3 950 eci um

Anzeigenpreise nach Preisliste Nr. 8, gültig ab 1. 10. 1987

Verlagsvertretungen Ausland (Bezugspreise in Klammern):

Belgien: Office International des Périodiques (O.I.P.), Avenue Marnix 30, B-1050 Brüssel (Einzelheft 182,– bfr, Jahresabonnement 1963,– bfr)

Dänemark: Harck + Gjellerups Booksellers Ltd., Fiolstraede 31–33, DK-1171 Kopenhagen K. (Jahresabonnement 316,– dkr)

Frankreich: Librairie Parisienne de la Radio, 43, rue de Dunkerque, F-75010 Paris

Luxemburg: Messageries Paul Kraus, 5, rue de Hollerich, Luxembourg

Niederlande: De Muiderkring BV, Electronics House, Postbus 313, 1380-AH Weesp (Einzelheft 8,50 hfl, Jahresabonnement 164,50 hfl)

Österreich: Erb-Verlag Ges.m.b.H. & Co., KG, Buch- u. Zeitschriftenvertrieb, Amerlingstr. 1, A-1061 Wien (Einzelheft 60,– öS, Jahresabonnement 620,– öS)

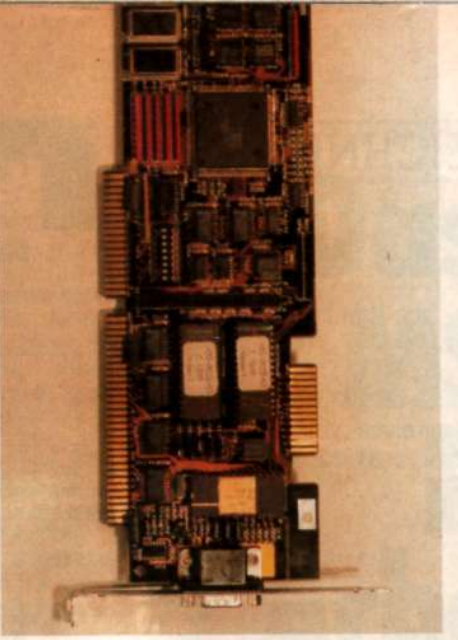
Schweiz: Verlag Thali AG, CH-6285 Hitzkirch (Luzern) (Einzelheft 7,– sfr, Jahresabonnement 67,20 sfr, je nach Kurssituation)

Urheberrechte: Die in der Zeitschrift veröffentlichten Beiträge sind urheberrechtlich geschützt. Für Bauanleitungen, Schaltungen und Programme zeichnen die Verfasser bzw. Entwickler verantwortlich; für Fehler im Text, in Schaltbildern, Aufbauskizzen, Programm-Listings usw. kann die Redaktion weder eine juristische Verantwortung noch irgendeine Haftung übernehmen.
Printed in Germany. Imprimé en Allemagne. © 1988 für alle Beiträge bei Franzis-Verlag GmbH

VORSCHAU

VGA für schnelle PC

Ein 16-Bit-Anschluß an den Systembus – und damit doppelt breite Zugriffe – sowie ein Bildspeicher mit 512 KByte Bildspeicher, für hohe Auslesegeschwindigkeit bestückt mit dual-ported-Video-RAM, sind nur zwei der besonderen Merkmale einer VGA, die wir in der nächsten Ausgabe vorstellen. Ein Vergleich mit anderen VGA-Karten und einige Grundlagen zum Thema VGA und den Techniken dieser Karte machen den neuen Grafikstandard transparent.



Star-Parade

In einem ausführlichen Test zeigten die Programme der Star-Division, daß auch preiswerte Programme für PC durchaus professionellen Ansprüchen gerecht werden können. Wir stellen eine Textverarbeitung, eine Tabellenkalkulation, eine Datenbank und eine nützliche MS-DOS-Oberfläche für jeden PC-Anwender vor.

Modula

In Ausgabe 10 gibt es wieder Futter für die Modula-Fans unter den mc-Lesern. Einmal einen Test der SPC-Modula-Version für den Atari ST, und dann beginnt eine Serie, die den Betriebssystembau mit Modula-2 beleuchtet. Denn Modula ist eine Programmiersprache gerade für die Systemprogrammierung.

Scanner im Überblick

In einer Marktübersicht können Sie anhand der Daten ermitteln, welcher Scanner für Ihre Bedürfnisse am besten zugeschnitten ist. Handfeste Tips für die Auswahl runden die Hilfestellung ab. Mit diesen Peripheriegeräten und der richtigen Software gestaltet sich die Arbeit am PC wesentlich effektiver.

Macintosh

Wir machen hiermit öffentlich bekannt, daß wir in der Redaktion einen Macintosh stehen haben. Für diesen schönen Computer gibt es in Ausgabe 10 ein Beispielprogramm, das das Typische der Programmierung des Macintosh an einem praktischen Beispiel zeigt. DFÜ-Unterstützung unter Anwendung der Resourcen des Mac. Also etwas Subtiles in der nächsten Ausgabe von mc.

Materialien zu mc

Rufen Sie 089/59 54 68 an, es meldet sich die Firma Shamrock, die in unserem Auftrag die Sammeldisketten für je 25 DM vertreibt:

MS-DOS-Sammeldisk 19: Turbo-Pascal ohne Absturz und Linien in C aus mc 1/88; GALs programmieren aus mc 1–2/88, Festplatte sicherer gemacht; Rentenberechnung; Batch-Dateien mit mehr Komfort und Bildschirm-Attribute in Turbo-Pascal aus mc 2/88.

MS-DOS-Sammeldisk 20: Pascal-Programm zur Simulation des Know-how-Computers aus mc 4/88.

MS-DOS-Sammeldisk 21: Parameterübergabe aus Turbo-Pascal aus mc 4/88, CGA-Emulation mit Hercules-Karte und schnelle SIN- und COS-Algorithmen aus mc 5/88.

MS-DOS-Sammeldisk 22: Programmieren unter OS/2, Assembler mit Struktur und VGA-Test aus mc 6/88; Modula-Programm zur Braitenberg-Welt aus mc 7/88.

MS-DOS-Sammeldisk 23: Alle mc-Jahresinhaltsverzeichnisse von 1981 bis 1988 als ASCII-Dateien, inklusive Suchprogramm.

ST-Sammeldisk 1: EPROM-Programmer und Taschenrechner aus mc 9/86 (TOS-Version vom 27. 11. 86 oder später nötig!)

ST-Sammeldisk 2: Grafikdemo und Minimon aus mc 5/88.

ST-Sammeldisk 3: 3-D-Darstellung von Funktionen in GFA-Basic aus mc 7/87, Assembler-Routinen in GFA-Basic und Programme zum Setzen des Uhrenbausteins DS1216E aus mc 9/87.

ST-Sammeldisk 4: Physik und Gehirn aus mc 9/87, Split-Speed und Co-Routinen in Modula-2 aus mc 11/87, div. Public-Domain-Programme (Achtung: zweiseitig formatierte Diskette!)

Das KI-Sonderheft von mc enthält kompaktes Wissen über das Gebiet der künstlichen Intelligenz. Lisp, Expertensysteme, KI-Programmiertechniken, Prolog, maschinelles Beweisen, Physik und Gehirn – das sind die wichtigsten Themen, die im Sonderheft zu finden sind. Es kostet 19.50 DM.

Ausgabe 10/88 erscheint am 26. Sept. 1988

MCI

PERSONAL COMPUTER

Bitte neueste Prospekte anfordern!

Auf alle Geräte 12 Monate Garantie. Preise gültig ab 1.12.87. Lieferbedingungen auf Anfrage. MCI MICRO COMPUTER INSTRUMENTS GMBH eingetragen AG Bergisch Gladbach HRB 2575 · Herstellung und Vertrieb von Microcomputern.

MCI

Bensberger Straße 252 · 5060 Bergisch Gladbach 2
Tel. (02202) 1080
Fax: (02202) 31009 · Telex: 8873518

Vergleichen, vergleichen, und nochmals vergleichen...

dann kaufen Sie Ihren HIGHSCREEN PC doch bei VOBIS!

Bei HIGHSCREEN PCs und ATs stimmen nicht nur die technischen Daten: auch die Komponenten, mit denen Sie am meisten in Berührung kommen, überzeugen: Monitor und Tastatur!

1.) Monitor HIGHSCREEN Z 15: 1 Zoll besser! 15" statt 14" große Bildröhre, leicht nachleuchtend (dadurch erhöhte Flimmerfreiheit), DUAL-Modus, d.h. kompatibel zum Monochrome **und** Color-Grafik-Adapter. Außerdem Schalter für automatische Invers-Darstellung (schwarz auf weiß). Und durch den 1 Zoll größeren Bildschirm ist weiterer Abstand von der Bildröhre möglich: Ihre Augen werden es Ihnen danken.

HIGHSCREEN Z 15

349.-

Einzelpreis

In Verbindung mit HIGHSCREEN TOWER PC und AT gegen Aufpreis von

279.-

im Gesamtpaket enthalten.

SE DESIGN AACHEN

NEUERÖFFNUNG:

Ab 3. September auch in Hamburg, Esplanade 41 (Finnlandhaus) Tel. 040/35 36 58

(Schon die 2. VOBIS-Filiale in Hamburg: Kompetent und preiswert setzt sich eben durch!)

TOWER	PC 2 LW 360 K	AT 286 1 LW 1.2 MB	AT 286-16 B 1 LW 1.2 MB	AT 386 1 LW 1.2 MB
Grundpreis	**1799.-**	**3195.-**	**4199.-**	**6499.-**
HighTech Tastatur	●	●	●	●
Z 15 Monitor 15" paperwhite	279.-	279.-	279.-	279.-
Alternativen:				
640 K Ram	●	●	–	–
1 MB Ram	–	400.-	●	–
2 MB Ram	–	–	500.-	●
4 MB Ram	–	–	1800.-	auf Anfrage
20 MB Platte	–	●	●	–
30 MB Platte	600.-	–	–	–
40 MB Platte	–	500.-	500.-	●
80 MB Platte	–	1000.-	1000.-	500.-

● im Grundpreis enthalten – nicht lieferbar

Preise verstehen sich als Aufpreise zum Grundpreis in Form eines Komplettpaketes.

2.) HIGHSCREEN HighTech-Tastatur:

Mit richtigen mechanischen Schaltern (sog. „Double Click").
Muß man einfach gesehen (und gefühlt) haben. Incl. abnehmbarer Staubschutzhaube.
Einzelpreis: **225.- DM** Bei allen HIGHSCREEN TOWER PCs und ATs im Gesamtpaket enthalten.

VOBIS MICROCOMPUTER

kompetent und preiswert

HAUPT-VERWALTUNG:
Postfach 1778
Rotter Bruch 32-34
5100 AACHEN
☎ 0241/50 00 81
Tx 832 389 vobis d

1000 BERLIN 30 Kurfürstenstr. 101 · 030/2 13 94 80
2000 HAMBURG Krohnskamp 15 · 040/2 79 46 76
2300 KIEL Sophienblatt 74-78 · 0431/67 86 22
2800 BREMEN Violenstraße 37 · 0421/32 04 20
3000 HANNOVER Berliner Allee 47 · 0511/81 65 71
4000 DÜSSELDORF Wielandstr. 21 · 0211/35 99 64
4100 DUISBURG 1 Friedrich-Wilhelm-Str. 30 · 0203/2 78 63
4150 KREFELD Ostwall 92 · 02151/80 07 93
4300 ESSEN Huyssenallee 3 · 0201/23 17 74
4600 DORTMUND Hamburger Str. 110 · 0231/57 30 72
4800 BIELEFELD Herforder Str. 106 · 0521/6 38 78
5000 KÖLN Mathiasstr. 24-26 · 0221/24 86 42
5100 AACHEN Viktorstr. 74 · 0241/54 31 00
Großhändlerstr. 60 · 0241/2 44 94
(gemeinsam mit Foto PORST)
6000 FRANKFURT Frankenallee 207/209 · 069/73 40 49
7000 STUTTGART Marienstr. 11-13 · 0711/60 63 36
6400 FULDA Mittelstr. 19/21 · 0661/7 82 66
(gemeinsam mit Foto PORST)
7500 KARLSRUHE Kriegsstr. 27/29 (am BGH) · 0721/37 82 68
7750 KONSTANZ Knnutlinger Str. 18 · 07531/1 55 60
8000 MÜNCHEN Aberlestr. 3 · 089/77 21 10
8500 NÜRNBERG Vordere Ledergasse 9 · 0911/23 29 95
8900 AUGSBURG Jakoberstr. 16 · 0821/152349

Deutschlands umsatzgrößter Microcomputer-Spezialist